



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROBOTICKÁ PALETIZACE ŘÍZENÁ Z PLC

ROBOTIC PALLETIZATION CONTROLLED BY PLC

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Adam Šefránek

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Petr Pivoňka, CSc.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Adam Šefránek

ID: 186202

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Robotická paletizace řízená z PLC

POKyny PRO VYPRACOVÁNÍ:

1. Provedte rešerši týkající se paletizace v průmyslu.
2. Popište způsob řízení ABB robota z PLC od firmy B&R Automatizace.
3. Navrhněte způsob automatického generování paletizačního vzoru ze vstupních parametrů.
4. Vytvořte digitální dvojče paletizačního pracoviště.
5. Naprogramujte paletizační pracoviště a ověřte jeho funkčnost.
6. Vytvořte vizualizaci umožňující zadávání parametrů pro paletizaci.

Práce bude realizována ve spolupráci se společností B&R automatizace, spol. s r.o.

DOPORUČENÁ LITERATURA:

[1] Firemní literatura B&R: Automation Studio V4

[2] Firemní literatura B&R: Produkty pro řízení pohybu a vizualizace společnosti B&R

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: prof. Ing. Petr Pivoňka, CSc.

Konzultant: Ing. Tomáš Matucha, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá robotickou paletizací řízenou z PLC. První část uvádí příklady paletizačních zařízení. Druhá část ukazuje způsob řízení ABB robota pomocí PLC společnosti B&R Automatizace. Třetí část se zabývá vybranými metodami pro automatický návrh paletizačních vzorů, které jsou ve čtvrté části implementovány v podobě knihovny pro PLC. Pátá část uvádí postup tvorby virtuálního paletizačního pracoviště, které je v závěrečné části použito pro ověření knihovny pro návrh paletizačních vzorů.

KLÍČOVÁ SLOVA

Robotická paletizace, paletizační vzory, B&R Automatizace, ROBOTemplate, PLC, virtuální pracoviště

ABSTRACT

The master's thesis deals with a robotic palletizing controlled by PLC. The first part provides examples of palletizing machines. The second part shows how ABB robot is controlled by B&R Automation PLC. The third part introduces selected approaches for automated pallet pattern generation. The fourth part designs library that allows pallet patterns to be generated using B&R Automation PLC. Fifth part covers creation of virtual palletizing cell, which is used to test library for automated pallet pattern generation.

KEYWORDS

Robotic palletizing, pallet patterns, B&R Automation, ROBOTemplate, PLC, digital twin

ŠEFRÁNEK, Adam. *Robotická paletizace řízená z PLC*. Brno, 2020, 84 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: prof. Ing. Petr Pivoňka, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Robotická paletizace řízená z PLC“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 1. 6. 2020

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce panu prof. Ing. Petru Pivoňkovi, CSc. za vedení a podnětné návrhy k práci. Děkuji společnosti B&R Automatizace za projevenou důvěru a panu Ing. Tomáši Matuchovi, Ph.D. za cenné připomínky.

Brno 1. 6. 2020

.....

podpis autora

Obsah

Úvod	11
1 Paletizační zařízení	12
1.1 Robotické paletizátory	13
1.1.1 Chapadla pro paletizaci	15
2 Řízení robota ABB	17
2.1 Dosavadní způsob řízení robota	17
2.1.1 RAPID programy	17
2.2 Řízení s využitím PLC	18
2.2.1 Komunikace	18
2.2.2 Řízení robota v automatickém režimu	21
2.2.3 Vytváření pohybových procedur	23
2.2.4 Úprava cílových poloh	25
2.2.5 Přenos souborů do kontroléru robota	25
3 Paletizační vzory	26
3.1 Zápis problému	26
3.2 Redukce množiny možných řešení	27
3.3 Bloková heuristika	27
3.3.1 Metoda se čtyřmi oblastmi	28
3.3.2 Metoda s pěti oblastmi	30
3.3.3 Rekurzivní metoda s pěti oblastmi	31
3.3.4 L-přístup	34
4 Implementace knihovny pro generování paletizačních vzorů	39
4.1 Koncepce	39
4.2 Názvosloví	39
4.3 Datový typ pro rozměry	40
4.4 Normalizované body	40
4.5 Určení optimálního rozmístění	43
4.6 Určení pozice krabic	44
4.7 Seřazení krabic	46
4.8 Funkční blok RTPalletBasic	47
4.9 Převod poloh krabic na polohy robota	48

5	Digitální dvojče	51
5.1	RobotStudio	51
5.2	Tvorba scény	52
5.3	Generování krabic	53
5.4	Dopravník	55
5.5	Simulace chapadla	55
5.6	Simulace palety	57
5.7	Propojení pracoviště s PLC	58
5.7.1	Konfigurace stanice v RobotStudios	60
5.7.2	Vytvoření virtuálních signálů	62
5.7.3	Propojení signálů kontroléru s inteligentními komponenty . . .	63
5.7.4	Vytvoření programu pro komunikaci	64
6	Použití knihovny pro generování vzorů	66
6.1	Funkční blok pro návrh paletizačního vzoru	66
6.2	Pracovní cyklus paletizačního pracoviště	66
6.3	Vizualizace v mapp View	68
6.3.1	Návrh rozmístění krabic na paletě	68
6.3.2	Ovládání paletizačního pracoviště	70
6.4	Ověření funkčnosti	71
	Závěr	74
	Literatura	75
A	Datové struktury funkčního bloku RBTemplateRobotCtrl	79
A.1	Datové struktury vstupních proměnných	79
A.2	Datové struktury výstupních proměnných	81
B	Obsah přiloženého CD	84

Seznam obrázků

1.1	Paletizační zařízení <i>Box Palletizer VPM-BL</i> společnosti Verbruggen [2].	13
1.2	Příklad paletizační buňky společnosti Robotic Automation P/L [6].	14
1.3	Standardní paletizační buňka společnosti Pan Pac Engineering a/s [7].	14
1.4	Paletizační buňka s automatickým dávkováním palet, dvěma nakládacími místy a ovinovacím strojem [8].	15
1.5	Často používané typy chapadel.	16
2.1	Kontrolér IRC5 a ovládací jednotka FlexPendant. Převzato z [12].	17
2.2	Koncept uložení a načítání programů [13].	18
2.3	Funkční blok RBTTemplateCommTcp.	19
2.4	Datové struktury přenášené mezi PLC a robotem.	20
2.5	Funkční blok RBTTemplateRobotCtrl.	21
3.1	Typy rozmístění v závislosti na typu řezů.	26
3.2	Ukázka vytvořeného rozmístění. Převzato z [18].	29
3.3	Příklad rozmístění s nevyužitou středovou oblastí. Převzato z [18].	29
3.4	Převzato z [19].	30
3.5	Rozdělení palety na pět oblastí podle [19].	30
3.6	Reprezentace rozdělení palety na pět oblastí podle [21].	32
3.7	Příklad stromové struktury.	33
3.8	Příklady symetrických rozdělení podle [20].	34
3.9	Grafické znázornění L oblasti.	35
3.10	Pět způsobů rozdělení L oblasti na dvě menší L oblasti.	36
3.11	Dva způsoby rozdělení R oblasti na dvě menší L oblasti.	36
3.12	Způsoby dělení L oblasti na dvě R oblasti.	38
4.1	Koncepce knihovny RTPallet.	39
4.2	Funkce RTPalletGridPoints.	41
4.3	Vývojový diagram funkce RTPalletGridPoints.	42
4.4	Funkce RTPalletFiveBlock.	43
4.5	Funkce RTPalletFiveBlockToBoxPos.	44
4.6	Význam proměnných v datovém typu RTPalletBoxPosType.	45
4.7	Možná pořadí nakládání krabic na paletu v rámci jedné vrstvy.	46
4.8	Funkce RTPalletSortBoxPos.	46
4.9	Funkční blok RTPalletBasic.	47
4.10	Funkce RTPalletPosToRobtarget.	49
4.11	Příklad natočení souřadného systému nástroje.	50
4.12	Dva způsoby uchopení krabice.	50
5.1	Ukázka prostředí RobotStudio.	51
5.2	Výsledná podoba virtuálního pracoviště, pohled 1.	52

5.3	Výsledná podoba virtuálního pracoviště, pohled 2.	53
5.4	Způsob realizace inteligentní komponenty pro generování krabic. . . .	54
5.5	Válečkový dopravník se dvěma snímači.	55
5.6	Parametry inteligentní komponenty pro vytvoření modelu chapadla. .	56
5.7	Způsob realizace inteligentní komponenty Gripper pro simulaci cha- padla.	57
5.8	Způsob propojení inteligentních komponent Gripper a PalletQueue. .	57
5.9	Znázornění způsobu přenosu signálů mezi PLC a RobotStudiem. . . .	59
5.10	Možné způsoby otevření okna pro konfiguraci rozšíření.	61
5.11	Okno pro konfiguraci rozšíření.	62
5.12	Okno pro správu signálů.	62
5.13	Parametry nového signálu.	63
5.14	Ukázka propojení signálů kontroléru s inteligentními komponenty. . .	64
5.15	Posloupnost provádění programu pro komunikaci s PLC.	64
6.1	Pracovní cyklus paletizačního pracoviště.	67
6.2	Obrazovky jsou v mapp View vytvářeny pomocí <i>widgetů</i> . Převzato z [28].	68
6.3	Stránka pro návrh rozmístění krabic na paletě.	69
6.4	Stránka ovládání paletizačního pracoviště.	70
6.5	Vygenerované rozmístění krabic v testovacích vrstvách.	72
6.6	Průběh paletizace.	73

Seznam tabulek

2.1	Parametry funkčního bloku RBTemplateCommTcp	19
2.2	Vybrané signály struktury ControllerInput.	22
2.3	Vybrané prvky struktury RobotInput.	22
2.4	Vybrané prvky struktury ControllerStatus.	23
2.5	Vybrané prvky struktury RobotStatus.	23
2.6	Příkazy funkčního bloku RBTemplateTeaching.	24
4.1	Rozměry palet podle ISO 6780:2003. Převzato z [24].	40
4.2	Prvky datového typu RTPalletFiveBlockReturnType	44
4.3	Prvky datového typu RTPalletBoxPosType	45
4.4	Parametry vstupu DrawLayer	48
5.1	Signály přenášené mezi PLC a virtuálním pracovištěm – příkazy. . . .	59
5.2	Signály přenášené mezi virtuálním pracovištěm a PLC – stavové signály.	60
6.1	Popis procedur paletizačního pracoviště	67
A.1	Prvky struktury RBTemplateRobotCtrlInputCmdType	79
A.2	Prvky struktury RBTemplateRobotCtrlInputRobType	80
A.3	Prvky struktury RBTemplateRobotCtrlStatusType	81
A.4	Prvky struktury RBTemplateRobotCtrlRobInfoType	82
A.5	Prvky struktury RBTemplateRobotMotionDataType	83

Úvod

Přeprava materiálu a zboží na paletách je v současnosti považována za standard. Nakládání na paletu provádějí výhradně automatizovaná či automatická zařízení. Jako příklady těchto zařízení lze uvést portálové manipulátory, rozsáhlé balicí linky a robotické paletizátory. Právě robotické paletizátory jsou stále více rozšířené, jelikož umožňují snadnější přizpůsobení než jednoúčelová, na míru navržená zařízení.

Použití robotů však vyžaduje jejich naprogramování pro danou činnost, což znamená, že je potřeba odborník znalý způsobu činnosti a programování použitého robota. Z toho důvodu přišla společnost B&R Automatizace s integrační platformou ROBOTemplate, která převádí programování a řízení robota do PLC. Jelikož v nedávné době došlo ke začlenění společnosti B&R Automatizace do skupiny ABB, je integrační platforma navržena právě pro roboty společnosti ABB. Ve výsledku stačí robota jednou naprogramovat připravenou šablonou, následné řízení robota, vytváření procedur, jejich úprava a spouštění probíhá z PLC, například pomocí vizualizace na dotykovém panelu. Tento způsob řešení přináší řadu výhod. Řízení pomocí systému B&R je mnohem jednodušší a uživatelsky přívětivější. Úpravy procesu může zákazník provádět svépomocí, čímž se snižuje potřebný čas a náklady spojené s přizpůsobením aplikace.

Byť je použití ROBOTemplate možné samostatně, je snaha zjednodušit způsob jeho nasazení sadou modulů pro konkrétní technologické procesy využívající robota. Může se jednat například o manipulaci s materiálem, lakování, svařování či obrábění. Moduly v zásadě vhodně transformují vstupní data, která se liší v závislosti na procesu, na polohy robota. Do jisté míry je tak automatizováno vytváření procedur, které zajišťují jeho činnost.

Jedním z takových procesů je robotická paletizace. Aby mohla být provedena, je nutné znát rozmístění zboží na paletě. Pro návrh rozmístění již existuje software – zde lze uvést pro představu uvést RobotStudio Palletizing PowerPac –, nicméně ten není určen pro běh v PLC. Vyvstává tedy otázka, jak přenést informace o rozmístění zboží na paletě do PLC tak, aby z něj následně mohla být řízena činnost robota pomocí ROBOTemplate.

Cílem této diplomové práce je vytvořit modul, který umožní automatický návrh paletizačního vzoru a připraví data potřebná pro provedení paletizace robotem. Modul bude možné spustit přímo z PLC společnosti B&R Automatizace a v součinnosti s ROBOTemplate umožní snadnou realizaci paletizace.

1 Paletizační zařízení

Paletizační zařízení – paletizátor – je zařízení, které slouží k nakládání zboží, nejčastěji krabic, přepravek či pytlů, na paletu. Nakládání probíhá po vrstvách. Zboží je v jednotlivých vrstvách umístěno podle předem známého uspořádání. Při návrhu vhodného uspořádání je nejčastěji snaha maximalizovat množství zboží a zachovat celkovou stabilitu nákladu.

Paletizační zařízení se dělí podle způsobu nakládání zboží na paletu [1]:

- **High-Level Palletizers.** Při nakládání se pohybuje paleta. Zboží je nad paletou zformováno do určeného rozložení. Hotová formace je přesunuta na paletu, která se následně posune o výšku jedné vrstvy dolů. Proces se opakuje, dokud není dosaženo požadovaného počtu vrstev.
- **Low-Level Palletizers.** Paleta je po celou dobu nakládání nehybná. Součástí zařízení je pohyblivá plošina. Nejdříve je na plošině vytvořeno požadované rozložení zboží, následně je plošina zvednuta nad úroveň poslední vrstvy na paletě a zboží je z plošiny na paletu přesunuto.
- **Robotic Palletizers.** Nakládání zboží na paletu zajišťuje robot s vhodným chapadlem. Oproti předchozím dvěma typům je nakládání obecně pomalejší, nicméně robotické paletizátory poskytují největší variabilitu (nakládání zboží různých velikostí, nakládání na více palet současně, snadnější změna paletizačního vzoru).
- **Specializované paletizátory**, které jsou navrženy pro konkrétní typ zboží (barely, lahve, pytle aj.).

Součástí paletizátorů jsou často stohovací zařízení pro uchovávání prázdných palet a ovinovací stroje pro ovíjení naložených palet stretch fólií. Příklad takového zařízení je uveden na obr. 1.1. Z hlediska dělení paletizátorů se jedná o *Low-Level* paletizátor.



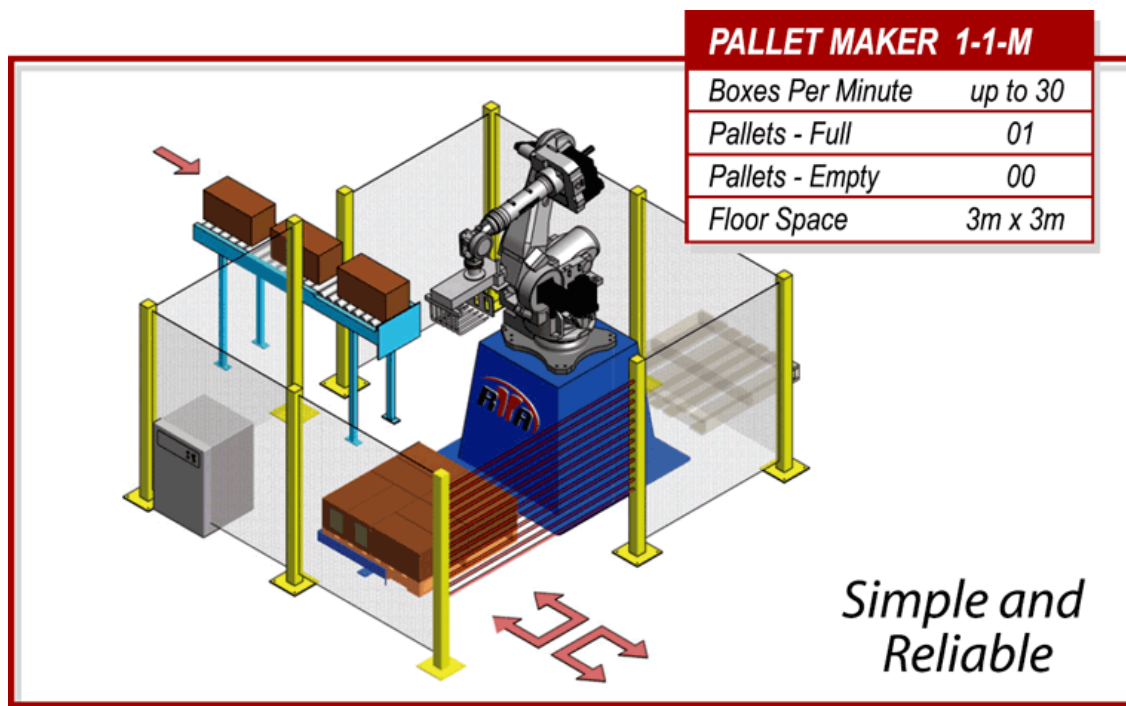
Obr. 1.1: Paletizační zařízení *Box Palletizer VPM-BL* společnosti Verbruggen [2].

1.1 Robotické paletizátory

Robotické paletizátory nacházejí svá uplatnění zejména u pomaleji běžících linek¹ nebo tam, kde je potřeba obsluhovat více míst nezávisle. Robot může navíc mezi vrstvy vkládat proložky a přenášet prázdné palety. Kromě toho lze použitím speciálních nástrojů – chapadel – manipulovat například s celými vrstvami [3].

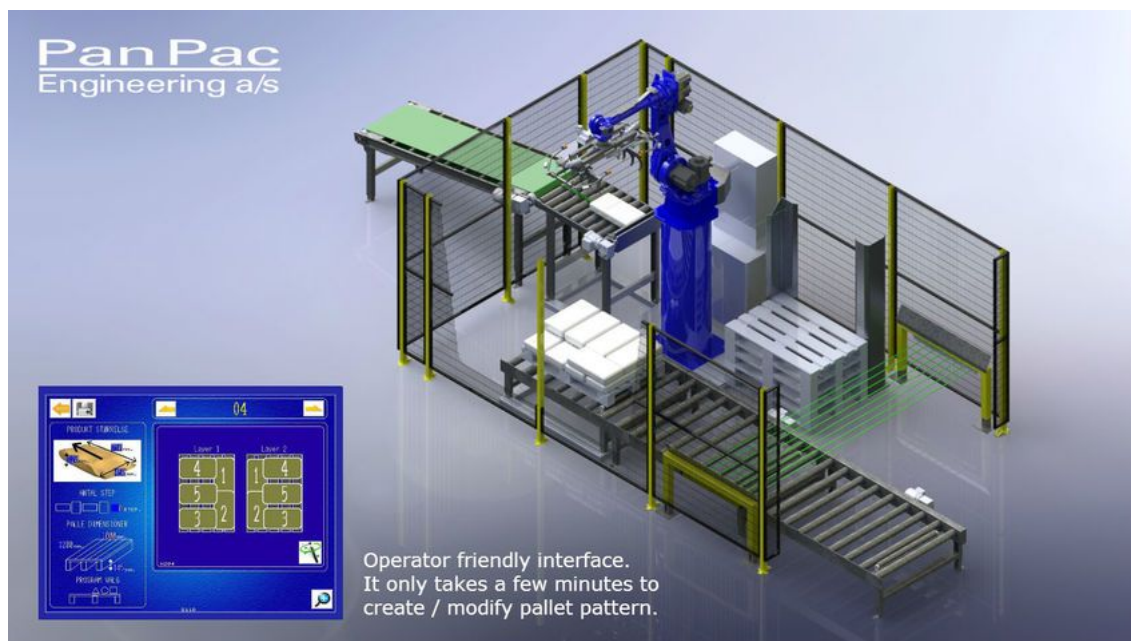
Mnozí výrobci nabízí již hotová řešení v podobě robotických paletizačních buněk. Příklady uvádí obr. 1.2 a 1.3. Buňky se v zásadě sestávají z robota s vhodným uchopovacím nástrojem, dopravníku pro přísun zboží, palety a bezpečnostních prvků, jako je oplocení či optické závory.

¹Rychlost samozřejmě závisí na konkrétní aplikaci. Srovnáním různých výrobců [3][4][5][6] lze říct, že rychlost se pohybuje mezi 10–30 produkty (krabice, pytle, přepravky aj.) za minutu.

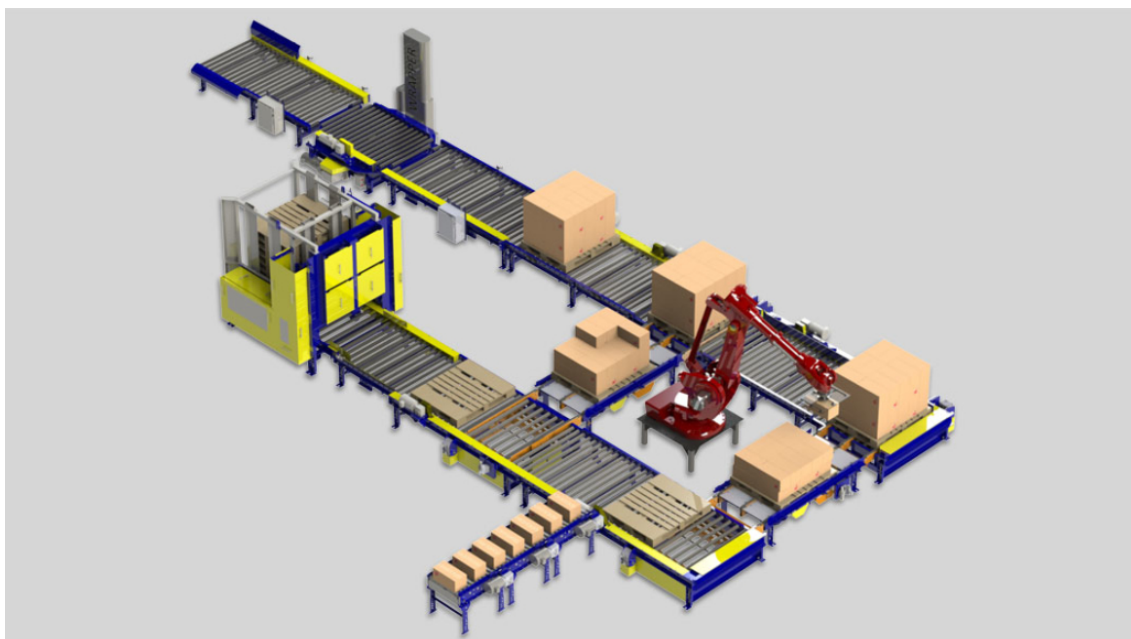


Obr. 1.2: Příklad paletizační buňky společnosti Robotic Automation P/L [6].

U složitějších zařízení bývá často součástí systém pro výměnu palet – dopravník, stohovač – a také ovinovací stroj. Příklad takového zařízení na obr. 1.4.



Obr. 1.3: Standardní paletizační buňka společnosti Pan Pac Engineering a/s [7].

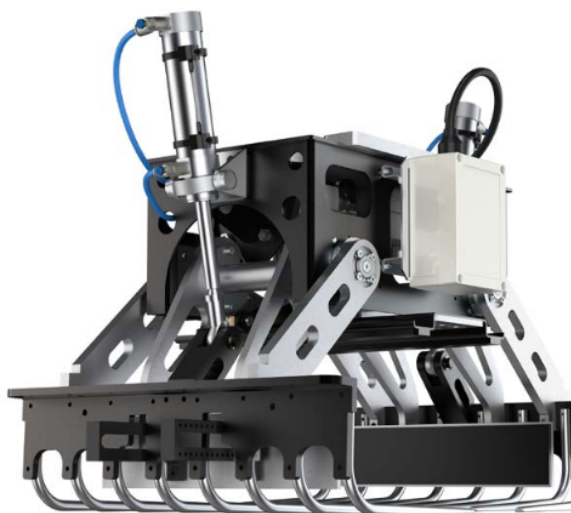


Obr. 1.4: Paletizační buňka s automatickým dávkováním palet, dvěma nakládacími místy a ovinovacím strojem [8].

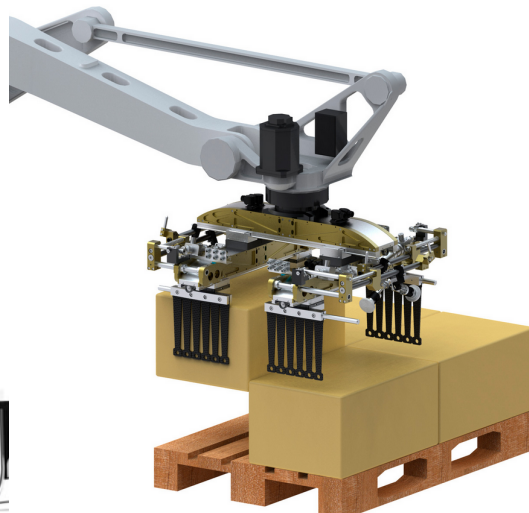
1.1.1 Chapadla pro paletizaci

Robot je univerzální stroj, který lze použít pro mnoho různých činností, například manipulaci s objekty, lakování či svařování. Aby však mohl konkrétní činnost vykonávat, je nutné jej vybavit vhodným nástrojem – efektozem. V oblasti paletizace se používají chapadla. Tato se liší konstrukcí a také principem, jakým je zboží uchopeno. Při volbě chapadla je potřeba brát v úvahu tvar a vlastnosti přenášeného předmětu.

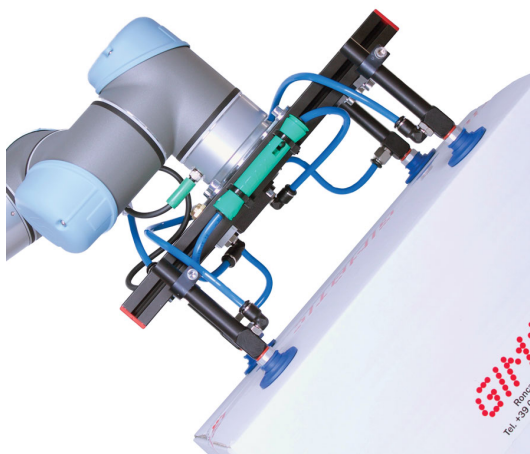
Z rozsáhlé nabídky chapadel lze vyčlenit tři, která jsou zřejmě nepoužívanější: čelisti pro manipulaci s pytli, čelisti pro manipulaci s krabicemi a vakuová chapadla. Ukázky vyjmenovaných typů jsou na obr. 1.5.



(a) čelisti pro manipulaci s pytli [9].



(b) čelisti pro přenášení krabic [10].



(c) Vakuové chapadlo [11].

Obr. 1.5: Často používané typy chapadel.

Při použití čelistí je nutné mít na zřeteli způsob jejich otevření a velikost v otevřeném stavu a naplánovat pořadí nakládání tak, aby bylo možné čelisti otevřít. V případě, že nelze prostor pro otevření čelistí zajistit, je možným řešením použití vakuových chapadel.

Základním prvkem vakuových chapadel jsou přísavky. Jejich počet a rozmístění lze určit podle potřeby. Přenášený předmět je přichycen shora, proto se vakuová chapadla nehodí pro těžké náklady – kartónové krabice s těžkým nákladem, papírové pytle s cementem aj. – , u kterých hrozí překročení pevnosti obalu a vysypání obsahu. Neumožňují rovněž přenášet otevřené předměty, například přepravky bez víka.

2 Řízení robota ABB

2.1 Dosavadní způsob řízení robota

K robotu se neodmyslitelně váže kontrolér, který zahrnuje všechny potřebné funkce, aby mohl robot vykonávat požadovanou činnost. Obsahuje silovou část pro řízení pohonů, výpočetní jednotku pro plánování trajektorie, vyhodnocuje bezpečnostní signály. Jeho prostřednictvím je naprogramována činnost robota. Současná verze kontroléru ABB nese označení IRC5.

Ke kontroléru je často připojen FlexPendant, ovládací jednotka s dotykovou obrazovkou pro operátora. Umožňuje například ručně pohybovat s osami robota (jogging), vytvářet, spouštět a upravovat procedury.



(a) Kontrolér IRC5.



(b) Ovládací jednotka FlexPendant.

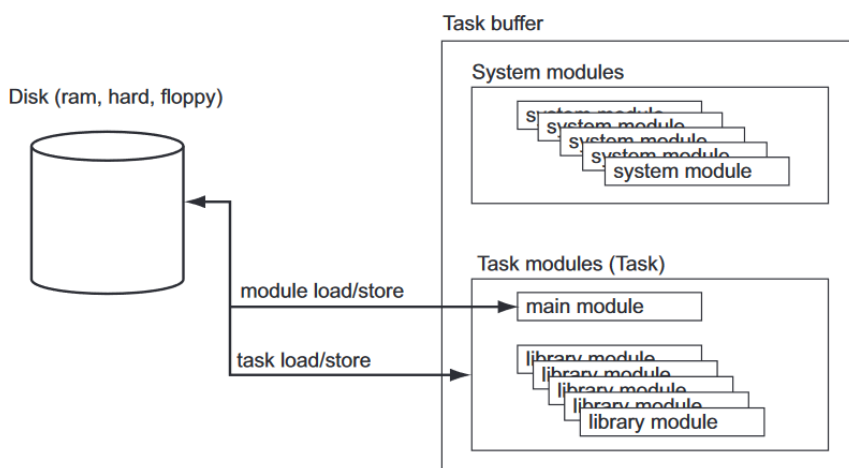
Obr. 2.1: Kontrolér IRC5 a ovládací jednotka FlexPendant. Převzato z [12].

2.1.1 RAPID programy

Programy robota jsou psány v jazyce RAPID. Program – task – se skládá z modulů. Modul může obsahovat typové definice, deklarace dat a rutin [13]. Rutiny jsou dále děleny na funkce, které mají návratovou hodnotu, procedury bez návratové hodnoty a tzv. trap rutiny, které se spustí automaticky při určitém přerušení.

Programování robota

Robota je možné programovat přes zmíněný FlexPendant, což může být zdlouhavé. Obrazovka je malá, dotyková klávesnice neumožňuje rychlé psaní. Tento způsob se



Obr. 2.2: Koncept uložení a načítání programů [13].

hodí spíše pro drobné úpravy. Dále je možné použít počítač a programovat robota přes prostředí RobotStudio, komplexní nástroj společnosti ABB.

2.2 Řízení s využitím PLC

RobotTEMPLATE je označení sady nástrojů pro řízení robota společnosti ABB pomocí PLC společnosti B&R Automation. Má za úkol zjednodušit a zrychlit vývoj nových zařízení s ABB robotem a B&R PLC. Základní verze zahrnuje

- komunikaci s robotem přes TCP/IP sockety,
- vytváření „uživatelských“ pohybových procedur z vyčítaných poloh robota,
- nahrávání souborů do paměti kontroléru robota přes Robot Web Services,
- ovládání motorů, načítání modulů a spouštění/zastavování procedur.

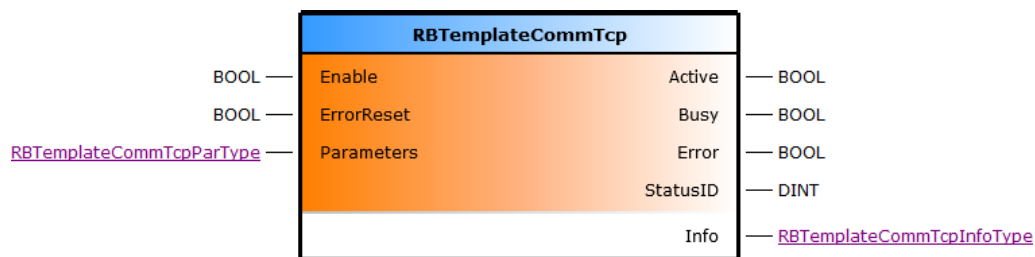
2.2.1 Komunikace

Vzhledem k požadavku na funkčnost knihovny RobotTEMPLATE v režimu simulace byla zvolena komunikace přes TCP/IP sockety. Jedno nebo obě zařízení – PLC, robot – nemusí být fyzicky přítomna, běží virtuálně. V tomto případě nelze snadno použít průmyslové sběrnice jako je například PROFINET.

V rámci komunikace přes TCP/IP sockety je robot v roli serveru, PLC představuje klienta. Výměna dat probíhá ve stylu požadavek–odpověď.

RBTemplateCommTcp

Ze strany PLC zajišťuje komunikaci funkční blok RBTemplateCommTcp. Ke své činnosti využívá funkčních bloků pro práci s TCP/IP sockety ze systémové knihovny AsTCP. Parametry funkčního bloku jsou uvedeny v tab. 2.1.



Obr. 2.3: Funkční blok RBTemplateCommTcp.

Výstup Active signalizuje, zda byl funkční blok inicializován a bylo navázáno spojení s robotem. Probíhající výměna dat, tj. odeslání dat z PLC a přijetí dat od robota, je signalizována hodnotou TRUE na výstupu Busy. Jakmile jsou data od robota zapsána do příslušné struktury, je výstup Busy nastaven na hodnotu FALSE až do dalšího volání funkčního bloku, kdy započne další komunikační cyklus.

Ve výstupní struktuře Info lze sledovat celkový počet odeslaných a přijatých dat od aktivace funkčního bloku a také dobu od odeslání do přijetí dat posledního komunikačního cyklu. Výstup Error signalizuje chybový stav, blíže je chyba specifikována kódem ve StatusID. Pro potvrzení chyby a obnovení komunikace slouží vstup ErrorReset.

Tab. 2.1: Parametry funkčního bloku RBTemplateCommTcp

Parametr	Datový typ	Popis
pRobotIpAdr	UDINT	ukazatel na IP adresu robota
RobotPort	UINT	port robota
PlcPort	UINT	port PLC
pSendData	UDINT	ukazatel na data k odeslání
SendDataLen	UDINT	délka odesílaných dat
pRecvData	UDINT	ukazatel na data k přijetí
RecvDataLen	UDINT	délka přijímaných dat
Timeout	UDINT	největší prodleva mezi odesláním a přijetím (ms)

Datové struktury

Přenášená data jsou rozdělena do tří samostatných portů (znázorněno na obr. 2.4). Jednotlivé porty jsou ze strany PLC obsluhovány instancemi funkčního bloku RBTemplateCommTcp.

Dvojice SysCmd, SysStatus slouží k ovládání robota. Struktura SysCmd obsahuje příkazy pro:

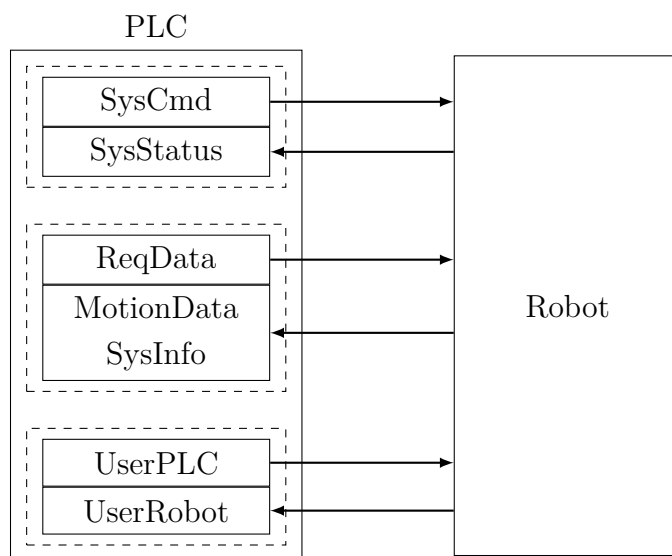
- zapnutí a vypnutí motorů,
- přesunutí program pointeru na začátek,
- spuštění pohybového tasku robota,
- zastavení vykonávání programu,
- načtení modulu,
- spuštění procedury.

Informace o stavu robota lze číst ve struktuře SysStatus. Jedná se o informace zdali

- jsou zapnuty/vypnuty motory,
- byl dokončen příkaz na přesunutí program pointeru,
- běží pohybový task robota,
- probíhá vykonávání procedury,
- bylo dokončeno načítání modulu,
- došlo k chybě.

Struktura ReqData umožňuje vybrat data, kterými má robot na daném portu odpovídat. SysInfo zahrnuje obecné informace, jako je například verze operačního systému nebo typové označení robota. Ke změně hodnot těchto dat dochází zřídka, na rozdíl od struktury MotionData, která obsahuje aktuální data o poloze a pohybu robota.

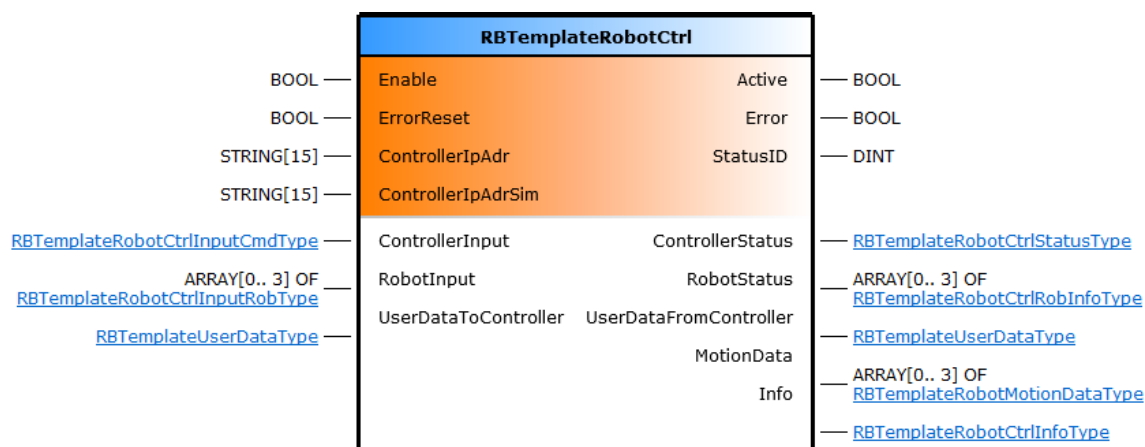
Poslední dvě struktury UserPLC a UserRobot jsou stejné. Obsahují několik proměnných, které lze v případě potřeby využít.



Obr. 2.4: Datové struktury přenášené mezi PLC a robotem.

2.2.2 Řízení robota v automatickém režimu

Pro řízení robota z PLC slouží funkční blok `RBTemplateRobotCtrl`, který zajišťuje komunikaci s robotem, hlídá stav robota, na jehož základě povoluje dostupné příkazy a ošetřuje chybové stavy. Funkční blok podporuje funkci `MultiMove`, tedy jednou instancí funkčního bloku `RBTemplateRobotCtrl` lze řídit až čtyři roboty připojené ke společnému kontroléru.



Obr. 2.5: Funkční blok `RBTemplateRobotCtrl`.

Celý funkční blok lze rozdělit na dvě hlavní části – komunikační a řídicí. Komunikační část zajišťuje komunikaci na třech portech s využitím funkčního bloku `RBTemplateCommTcp`. Po aktivaci funkční blok nejdříve ověří, zdali je kontrolér se zadanou IP adresou v síti. Následně se s kontrolérem pokusí navázat komunikaci. Výstup `Active` signalizuje, zda byl funkční blok inicializován a komunikace je aktivní. Pokud z nějakého důvodu není možné komunikaci navázat, řídicí část se neuplatní a na výstupu funkčního bloku je signalizována chyba.

V řídicí části se z příchozích stavových signálů ve struktuře `SystemStatus` zjistí režim kontroléru. Řízení robota je umožněno pouze v automatickém režimu. Dále je činnost řízena na základě vstupů funkčního bloku. Struktura `ControllerInput` (tab. 2.2) obsahuje signály pro řízení kontroléru, respektive všech robotů společně. Například aktivací vstupu `MotorsOn` jsou zapnuty motory všech připojených robotů, aktivací vstupu `StartAtMain` je spuštěn hlavní program každého robota.

Tab. 2.2: Vybrané signály struktury ControllerInput.

Název	Datový typ	Popis
MotorsOff	BOOL	zapnutí motorů
MotorsOn	BOOL	vypnutí motorů
StartAtMain	BOOL	spuštění hlavního programu
Stop	BOOL	zastavení vykonávání programu (procedury)

Pokud má robot zapnuty motory a běží jeho hlavní program, lze jej individuálně ovládat prostřednictvím vstupů ve struktuře RobotInput (tab. 2.3). Zde jsou vstupy zaměřeny na ovládání pohybu robota. Je možné například načítat moduly a spouštět procedury se zadaným názvem nebo omezit rychlost pohybu parametrem SpeedOverride.

Tab. 2.3: Vybrané prvky struktury RobotInput.

Název	Datový typ	Popis
ModuleSrcPath	STRING	cesta k souboru, který má být přenesen do kontroléru
ModuleName	STRING	název modulu, který má být přenesen do kontroléru
TransferModuleToController	BOOL	přenesení souboru do kontroléru
LoadModule	BOOL	načtení modulu do paměti kontroléru
ModuleName	STRING	název procedury, který má být spuštěna
StartProcedure	BOOL	spuštění vybrané procedury
SpeedOverride	USINT	omezení rychlosti robota (0–100 %)

Na výstup bloku jsou ve struktuře ControllerStatus (tab. 2.4) vyvedeny informace o stavu kontroléru a ve struktuře RobotStatus (tab. 2.5) stavové signály každého robota. Prostřednictvím těchto struktur je signalizován pohyb robota, průběh vykonávání procedury a dokončení zadaných příkazů (zapnutí motorů, spuštění programu, načtení modulu atd.). Struktura MotionData obsahuje informace o poloze a pohybu každého robota.

Úplný výčet všech výše vyjmenovaných struktur je možné nalézt v příloze A.

Tab. 2.4: Vybrané prvky struktury ControllerStatus.

Název	Datový typ	Popis
Ping	BOOL	robot je dostupný v síti
Connected	BOOL	komunikace s kontrolérem aktivní
AutomatOn	BOOL	kontrolér je v manuálním režimu
MotorsOn	BOOL	motory jsou zapnuty
MotorsOff	BOOL	motory jsou vypnuty
ExecutingMotionTask	BOOL	pohybový task robota běží/neběží
EmStop	BOOL	nouzové zastavení robota bylo aktivováno

Tab. 2.5: Vybrané prvky struktury RobotStatus.

Název	Datový typ	Popis
Active	BOOL	robot je připojen ke kontroléru
Error	BOOL	signalizace chyby
ErrorCode	BOOL	kód chyby
LimitSpeed	BOOL	omezení rychlosti pohybu aktivní
TransferDone	BOOL	soubor přenesen z PLC do kontroléru
LoadModuleDone	BOOL	načtení modulu dokončeno
ProcExecuting	BOOL	probíhá vykonávání procedury

Ošetření chyb

Chyby, které mohou nastat, jsou prakticky dvojího charakteru. Buď se jedná o chyby vzniklé voláním dalších funkčních bloků a funkcí, například nemožnost navázat spojení se projeví chybou funkčního bloku RBTemplateCommTcp, nebo o chyby vzniklé při samotném řízení. Zde lze uvést například chybu režimu kontroléru (manuální) nebo přerušení některého bezpečnostního okruhu robota.

V obou případech je nastalá chyba indikována výstupem Error a příslušným kódem StatusID. Potvrzení chyby, resp. obnovení činnosti funkčního bloku se provádí vstupem ErrorReset.

2.2.3 Vytváření pohybových procedur

Vytváření – učení – pohybových procedur je spojeno s manuálním režimem robota, kdy je umožněn pohyb os skrze FlexPendant. Po celou dobu učení probíhá přenos informací o poloze a na žádost je aktuální poloha zaznamenána v PLC. Takto lze

postupně projít celou trasu – výslednou pohybovou proceduru – a uložit potřebný počet poloh. Každé poloze jsou pak přiřazeny parametry, např. typ pohybu, rychlost, zóna, souřadný systém. Nakonec je vytvořen soubor s pohybovými instrukcemi v jazyce RAPID.

Funkční blok **RBTemplateTeaching**

Popsanou funkcionalitu vytváření pohybových procedur zajišťuje funkční blok **RBTemplateTeaching**. Komunikace s robotem je realizována opět funkčním blokem **RBTemplateCommTcp**.

Jedním ze vstupních parametrů funkčního bloku je ukazatel na seznam instrukcí a jeho délka. Každá položka v seznamu představuje jednu pohybovou instrukci pro robota a obsahuje všechny potřebné informace k jejímu vytvoření. Se seznamem instrukcí pracují příkazy uvedené v tab. 2.6.

Tab. 2.6: Příkazy funkčního bloku **RBTemplateTeaching**.

Proměnná	Datový typ	Popis
CreateInstruction	BOOL	vytvoří novou instrukci na konci seznamu
EditInstruction	BOOL	upraví instrukci s uvedeným indexem
InsertInstruction	BOOL	vloží instrukci na uvedené místo v seznamu
RemoveInstruction	BOOL	smaže instrukci s uvedeným indexem
CreateFile	BOOL	vytvoří soubor s pohybovou procedurou

Funkční blok **RBTemplateModGen**

Jakmile je seznam instrukcí připraven, je příkazem **CreateFile** předán funkčnímu bloku **RBTemplateModGen**, který jej převede na pohybové instrukce v jazyce RAPID a uloží do souboru s příponou **mod**.

Obecně se **mod** soubory skládají vždy z jednoho modulu, jehož název je stejný jako název souboru, a libovolného počtu pohybových procedur s unikátním názvem. Modulový soubor vytvořený funkčním blokem **RBTemplateModGen** obsahuje pouze jednu pohybovou proceduru, jejíž název je stejný jako název modulu doplněný o předponu „r“ (jako rutina, předpona „p“ se používá pro názvy poloh – **robtarget**, **jointtarget**).

Uvedený přístup zjednodušuje použití generovaných modulových souborů. Vzhledem k tomu, že vygenerovaný soubor obsahuje pouze jednu proceduru s přesně definovaným názvem, není nutné udržovat navíc informace o tom, jaké procedury jsou obsaženy v jednotlivých modulech. Případnou změnu procedury lze navíc provést přepsáním celého modulového souboru, aniž by došlo ke ztrátě jiných procedur.

2.2.4 Úprava cílových poloh

Instrukce MoveL a MoveJ v procedurách obsahují cílový bod pohybu. Ten může být uveden buď bezprostředně za instrukcí, nebo je možné jej definovat jako konstantu s unikátním názvem na začátku modulového souboru.

Druhá možnost je výhodná zejména v případě, pokud je potřeba cílové polohy dynamicky měnit. Například při robotickém odebrání výrobků z pásu kamera snímá odběrné místo a do PLC posílá údaje o polohách detekovaných výrobků. Následně je v modulu s procedurou pro přesunutí robota, resp. kleštin do polohy pro uchopení výrobku upraven pojmenovaný cílový bod. Takto upravený modul je přenesen do kontroléru robota, nahrán a spuštěn.

Pro úpravu cílových bodů slouží funkční blok RBTemplateTargetEditor. Umožňuje číst a upravovat body typu `robtarg`et i `jointtarg`et.

2.2.5 Přenos souborů do kontroléru robota

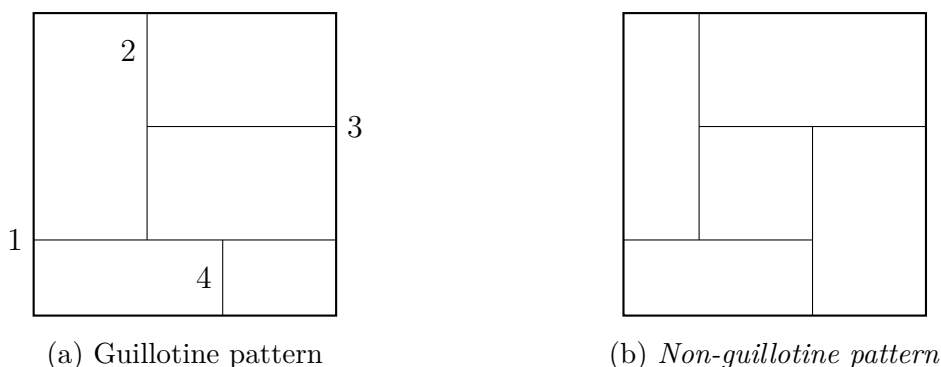
Soubory s moduly a procedurami mohou být do kontroléru robota přeneseny prostřednictvím FTP. Robot i PLC tuto variantu umožňují, nicméně u robota se jedná o placenou nástavbu kontroléru. Z toho důvodu bylo přistoupeno na řešení skrze tzv. Robot Web Services, webovou službu, která umožňuje užitím HTTP požadavků ovládat robota.

Komunikaci s kontrolérem prostřednictvím HTTP požadavků zajišťuje ze strany PLC funkční blok RBTemplateRestApi. Funkční blok umožňuje kromě nahrávání souboru do kontroléru rovněž soubory z kontroléru přenášet do PLC.

3 Paletizační vzory

Na problém generování paletizačního vzoru se lze dívat jako na speciální případ dvou-rozměrného problému řezání tvarů, přičemž všechny vyřezávané tvary jsou stejné. Na obdélníkovou plochu – paletu – o rozměrech (L, W) je potřeba umístit co největší počet menších obdélníkových ploch – krabic – o velikosti (l, w) . Jednotlivé menší plochy se samozřejmě nemohou překrývat a nesmí přesahovat okraje palety. Mohou být otočeny o 90° a musí být umístěny tak, aby všechny jejich hrany byly kolmé alespoň na některou z hran palety, tj. ortogonálně.

Literatura zabývající se problémem řezání zásob rozlišuje dva typy řezů. Guillotine cut označuje řez, který probíhá bez přerušení mezi dvěma protějšími hranami a dělí řezaný objekt na dvě samostatné části. Funguje tak například páková řezačka na papír nebo hydraulické nůžky na plech. V ostatních případech se řez nazývá *non-guillotine cut* [20]. Podoba vzniklého vzoru závisí na typu řezů. Obr. 3.1 ukazuje oba zástupce, čísla u obr. 3.1a značí možné pořadí vykonaných řezů v případě řezání. Praktická zkušenost dokazuje, že lepšího využití plochy – ať už při skutečném řezání nebo generování paletizačního vzoru – lze dosáhnout právě typem non-guillotine [18][19].



Obr. 3.1: Typy rozmístění v závislosti na typu řezů.

3.1 Zápis problému

Jak již bylo zmíněno, problém spočívá v optimálním rozmístění obdélníkových krabic o délce l a šířce w na obdélníkovou paletu o délce L a šířce W . Způsob zápisu problému uvádí například [21].

(L, W) jsou rozměry palety, (l, w) rozměry krabice, přičemž platí $L \geq W$, $l \geq w$. Hodnoty jsou přirozená čísla. Problém umístění N krabic lze popsat čtveřicí

(L, W, l, w) . Možné řešení pak jako (x_i, y_i, o_i) , $i = 1, \dots, N$, kde dvojice (x_i, y_i) odpovídá poloze levého spodního rohu i . krabice a o_i její orientaci (neotočená, otočená o 90°).

3.2 Redukce množiny možných řešení

Bez ohledu na použitý algoritmus je procházení všech možných umístění výpočetně náročné. Způsob, jak množinu možných řešení redukovat, aniž by došlo ke ztrátě některého z řešení, spočívá v definování specifických – alokačních – bodů, ve kterých mohou být krabice umístěny. Napříč literaturou se v zásadě používají dva přístupy.

První, kdy polohy (x_i, y_i) , $i = 1, \dots, N$ mohou nabývat pouze hodnot, které jsou lineární kombinací rozměrů (l, w) krabice. Množiny

$$\begin{aligned} S_L &= \{x \in \mathbb{Z}_0^+ \mid x = rl + sw, 0 \leq x \leq L, r, s \in \mathbb{Z}_0^+\} \\ S_W &= \{y \in \mathbb{Z}_0^+ \mid y = tl + uw, 0 \leq y \leq W, t, u \in \mathbb{Z}_0^+\} \end{aligned} \quad (3.1)$$

se nazývají normalizované body¹. \mathbb{Z}_0^+ je obor celých kladných čísel včetně 0. Druhý přístup omezuje polohy krabic na množiny R_L a R_W , kde

$$\begin{aligned} R_L &= \{x \in \mathbb{Z}_0^+ \mid y = \langle L - \hat{x} \rangle_{S_L}, \exists \hat{x} \in S_L\} \\ R_W &= \{y \in \mathbb{Z}_0^+ \mid y = \langle W - \hat{y} \rangle_{S_W}, \exists \hat{y} \in S_W\} \end{aligned} \quad (3.2)$$

a

$$\begin{aligned} \langle \hat{x} \rangle_{S_L} &= \max \{x \in S_L \mid x \leq \hat{x}\} \\ \langle \hat{y} \rangle_{S_W} &= \max \{y \in S_W \mid y \leq \hat{y}\}. \end{aligned} \quad (3.3)$$

Výše uvedené rovnice se shodují s rovnicemi uvedenými v [21], kde je rovněž možné nalézt důkazy, že při použití normalizovaných nebo rastrových bodů nedochází k újmě na obecnosti řešení.

3.3 Bloková heuristika

Z důvodu výpočetní náročnosti exaktních algoritmů jsou častěji využívány heuristické přístupy. Jedním z nich je tzv. bloková heuristika, která využívá dělení palety na menší oblasti. V následujících podkapitolách bude uveden základní princip několika z nich.

¹normal points, grid points

3.3.1 Metoda se čtyřmi oblastmi

Princip této metody popisuje článek [18]. Paletizační vzor je vytvořen rozdělením palety na nejvýše čtyři pomyslné pravoúhlé oblasti. Každá oblast obsahuje krabice uložené se stejnou orientací. Rozdělení palety probíhá ve dvou krocích.

V prvním kroku se hledá vhodné rozmístění krabic po obvodu palety. Cílem je využít co největší část obvodu². Lze definovat

$$F_n(S_n) = \max\{X_n \cdot l + Y_n \cdot w + F_{n-1}(S_{n-1})\} \quad (3.4)$$

za podmínky

$$X_n \cdot l + Y_n \cdot w \leq D_n, \quad n = 1, \dots, 4, \quad (3.5)$$

kde

$F_n(S_n)$ je největší součet délek hran umístěných krabic pro danou hranu n palety při počátečních podmínkách S_n ,

X_n počet krabic o délce l položených na délku podél hrany n palety,

Y_n počet krabic o šířce w položených na šířku podél hrany n palety,

D_n velikost dané hrany n palety,

S_n počáteční podmínka pro danou hranu n palety, může nabývat tří hodnot:

$$S_n = 1 : \quad X_n = 0, Y_n = 2,$$

$$S_n = 2 : \quad X_n = 2, Y_n = 0,$$

$$S_n = 3 : \quad X_n = 1, Y_n = 1.$$

Optimální hodnoty X_n a Y_n tvoří základ pomyslných oblastí. Každá oblast je umístěna v jednom z vrcholů palety a směřuje k jejímu středu. Orientace krabic v jednotlivých oblastech je předem určena a nelze ji v průběhu změnit. Oblast B_1 je tvořena z X_1 a Y_4 ; B_2 z X_2 a Y_1 ; B_3 z X_3 a Y_2 ; B_4 z X_4 a Y_3 .

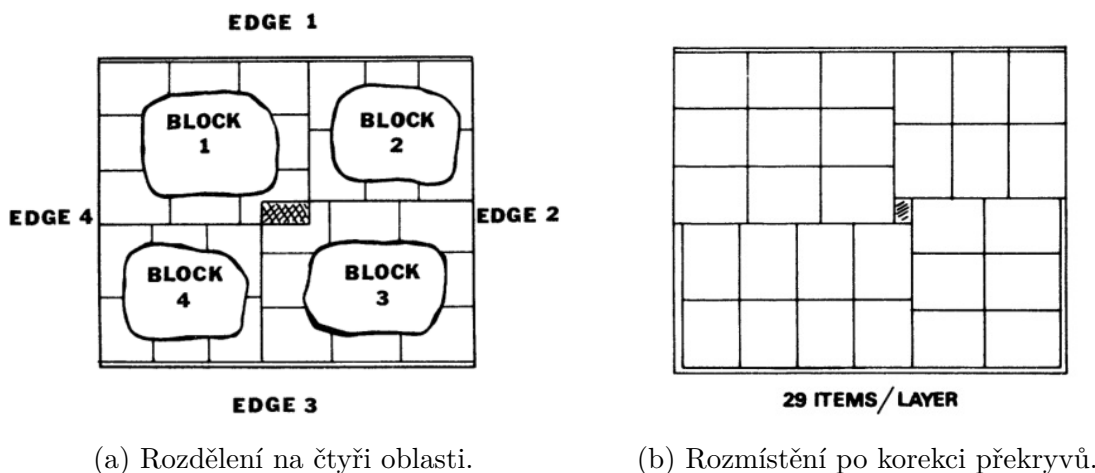
Například pokud $X_1 \geq 1$ a tedy $Y_4 \geq 1$, pak oblast B_1 existuje a obsahuje krabice orientované tak, že delší hranou (o rozměru l) leží podél hrany 1 palety a kratší hranou (o rozměru w) zcela logicky podél hrany 2 palety. Obr. 3.2a zobrazuje příklad rozmístění po prvním kroku včetně označených oblastí (block) a hran (edge).

Ve druhém kroku se v každé oblasti doplní krabice směrem ke středu palety tak, aby byly oblasti vyplněny³. Mohou nastat dvě problémové situace, které je potřeba dále řešit: oblasti se překrývají – případ vyznačen šrafováním na obr. 3.2a. –, nebo mezi nimi naopak zbývá dostatečné místo pro další krabice. Překrytí mezi oblastmi B_1 a B_3 nastává pokud

$$(D_1 - X_1 \cdot l) < X_3 \cdot l \quad \wedge \quad (D_4 - Y_4 \cdot w) < Y_2 \cdot w \quad (3.6)$$

²tzv. perimeter utilization

³v článku [18] je tento krok nazván jako inward projection

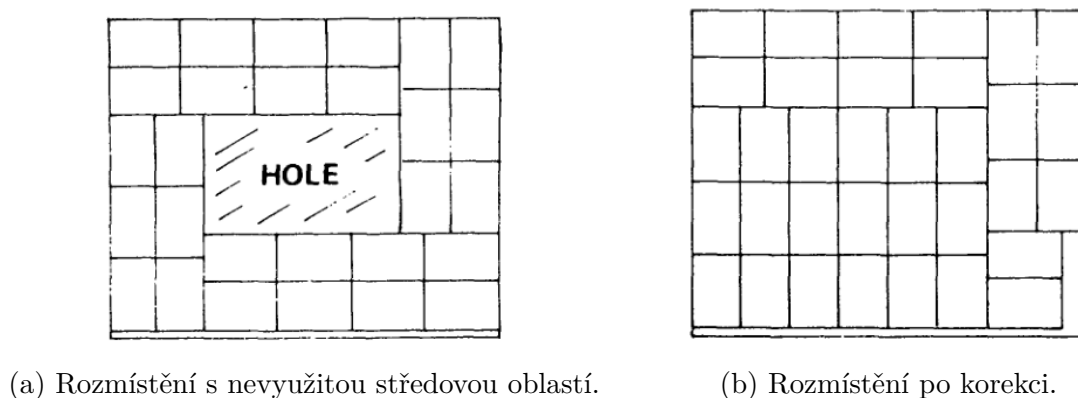


Obr. 3.2: Ukázka vytvořeného rozmístění. Převzato z [18].

a obdobně mezi oblastmi B_2 a B_4 pokud

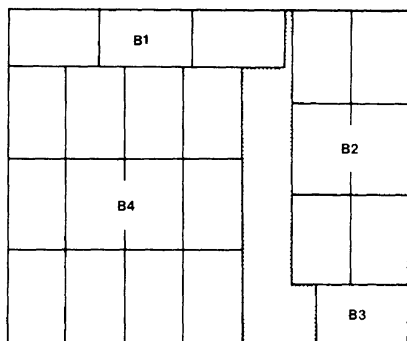
$$(D_1 - Y_1 \cdot w) < Y_3 \cdot w \quad \wedge \quad (D_2 - X_2 \cdot l) < X_4 \cdot l \quad (3.7)$$

Nastalé překrytí se „jednoduše“ odstraní zafixováním oblastí B_1 a B_2 a úpravou oblastí B_3 a B_4 změnou hodnot X_3 , Y_2 , X_4 , Y_3 . Pro příklad na obr. 3.2a by řešení spočívalo ve snížení hodnoty X_3 o 1 nebo ve zvýšení hodnoty Y_2 o 1.

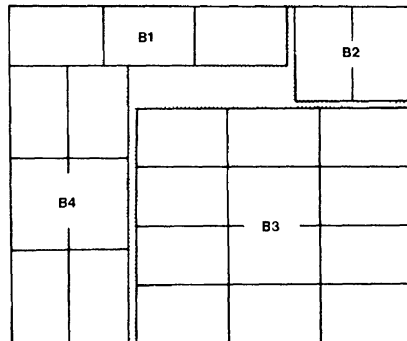


Obr. 3.3: Příklad rozmístění s nevyužitou středovou oblastí. Převzato z [18].

Nevýhoda výše popsaného přístupu spočívá ve způsobu řešení nevyužitého místa ve středové oblasti. Algoritmus zafixuje oblasti B_1 a B_2 a dále pracuje pouze s oblastmi B_3 a B_4 . Výsledkem může být například rozmístění na obr. 3.4a. Obr. 3.4b ukazuje optimálnější řešení, které využívá větší plochu palety, avšak nelze jej získat popsaným přístupem, jelikož vyžaduje variaci více oblastí.



(a) Rozmístění navržené Steudelovým přístupem.

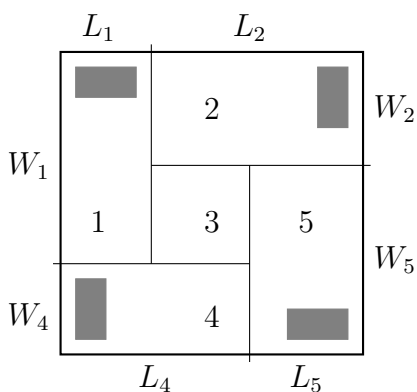


(b) Optimálnější rozmístění.

Obr. 3.4: Převzato z [19].

3.3.2 Metoda s pěti oblastmi

Metoda prezentovaná v článku [19] navazuje na metodu čtyřblovou. Zásadní změna spočívá v rozdělení palety na pět oblastí tak, jak je ukázáno na obr. 3.5. Na obrázku je rovněž šedými obdélníky zobrazena předem určená a neměnná orientace krabic v oblastech 1, 2, 4 a 5. Orientace krabic ve středové oblasti 3 není předem dána, protože závisí na velikostech ostatních oblastí. Algoritmus hledá nejvhodnější kombinace vnějších rozměrů L_1, L_2, L_4, L_5 a W_1, W_2, W_4, W_5 , tj. řešení s největším počtem umístěných krabic.



Obr. 3.5: Rozdělení palety na pět oblastí podle [19].

Vnější rozměry mohou nabývat pouze hodnot z množin P a Q .

$$\begin{aligned} P &= \{(r, s) \mid L - w < rl + sw \leq L, \ r, s \in \mathbb{Z}_+\} \\ Q &= \{(t, l) \mid W - w < tl + ul \leq W, \ t, l \in \mathbb{Z}_+\} \end{aligned} \quad (3.8)$$

S ohledem na naznačenou orientaci krabic jsou vnější rozměry definovány násled-

dovně:

$$\begin{aligned} L_1 &= sw, W_1 = tl; L_2 = rl, W_2 = uw; \\ L_4 &= rl, W_4 = uw; L_5 = sw, W_5 = tl. \end{aligned} \quad (3.9)$$

Metoda nemusí vždy vést k proveditelnému řešení. Navržené rozmístění může obsahovat vzájemně se překrývající oblasti. Překryv nastává, pokud platí podmínky

$$L - L_1 - L_5 < 0 \quad \wedge \quad W - W_1 - W_5 < 0 \quad (3.10)$$

nebo

$$L - L_2 - L_4 < 0 \quad \wedge \quad W - W_2 - W_4 < 0 \quad (3.11)$$

Kontrolou těchto podmínek se rozmístění s překrývajícími se oblastmi z řešení vyloučí. Záměna párů na protějších stranách je stejná jako otočení palety o 180° . Tím se množina rozmístění k analýze zmenšuje na polovinu. Pro každé analyzované rozmístění se určí počet umístěných krabic z .

$$z = \sum_{i=1}^5 z_i \quad (3.12)$$

$$z_i = \lfloor L_i/w \rfloor \cdot \lfloor W_i/l \rfloor, \quad i = 1, 5 \quad (3.13)$$

$$z_i = \lfloor L_i/l \rfloor \cdot \lfloor W_i/w \rfloor, \quad i = 2, 4 \quad (3.14)$$

$$z_3 = \max\{\lfloor L_3/l \rfloor \cdot \lfloor L_3/w \rfloor, \lfloor W_3/l \rfloor \cdot \lfloor W_3/w \rfloor\} \quad (3.15)$$

Hledá se řešení s největším počtem umístěných krabic z . Algoritmus může být předčasně ukončen, je-li nalezeno řešení, u kterého je zbývající místo na paletě menší než plocha vyžadována pro umístění jedné krabice.

3.3.3 Rekurzivní metoda s pěti oblastmi

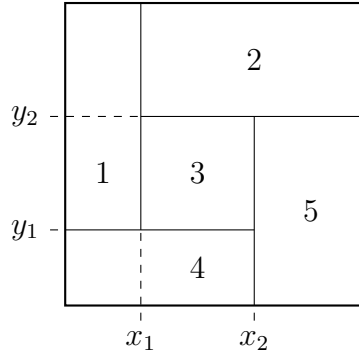
Autoři článku [20] provedli úpravy původní pětiblokové metody (kap. 3.3.2) na rekurzivní variantu. Ukazují, že v případě rekurzivního algoritmu lze optimálnější řešení dosáhnout nahrazením množin P a Q (rov. 3.8) množinami S_L, S_R podle rov. 3.1, případně množinami R_L, R_W podle rov. 3.2.

Reprezentace rozdělení

Z hlediska algoritmizace může být vhodné převést reprezentaci rozdělení na pět oblastí podle obr. 3.5 na reprezentaci podle obr. 3.6. Tento způsob používá kromě [20]

také [21]. Platí přepočet na vnější rozměry

$$\begin{aligned} L_1 &= x_1, \quad W_1 = W - y_1; \quad L_2 = L - x_1, \quad W_2 = W - y_2; \\ L_4 &= x_2, \quad W_4 = y_1; \quad L_5 = L - x_2, \quad W_5 = y_2; \end{aligned} \quad (3.16)$$



Obr. 3.6: Reprezentace rozdělení palety na pět oblastí podle [21].

Zavedení mezí

Zavedení dolní meze z_{lb} a horní meze z_{ub} má za cíl snížit výpočetní náročnost algoritmu. Meze jsou definovány následovně.

$$z_{lb} = \max\{\lfloor L/l \rfloor \lfloor W/w \rfloor, \lfloor L/w \rfloor \lfloor W/l \rfloor\} \quad (3.17)$$

$$z_{ub} = \lfloor L^*W^*/(lw) \rfloor, \quad (3.18)$$

kde

$$\begin{aligned} L^* &= w + \max\{x \mid x \in S_L\} \\ W^* &= w + \max\{y \mid y \in S_W\}. \end{aligned} \quad (3.19)$$

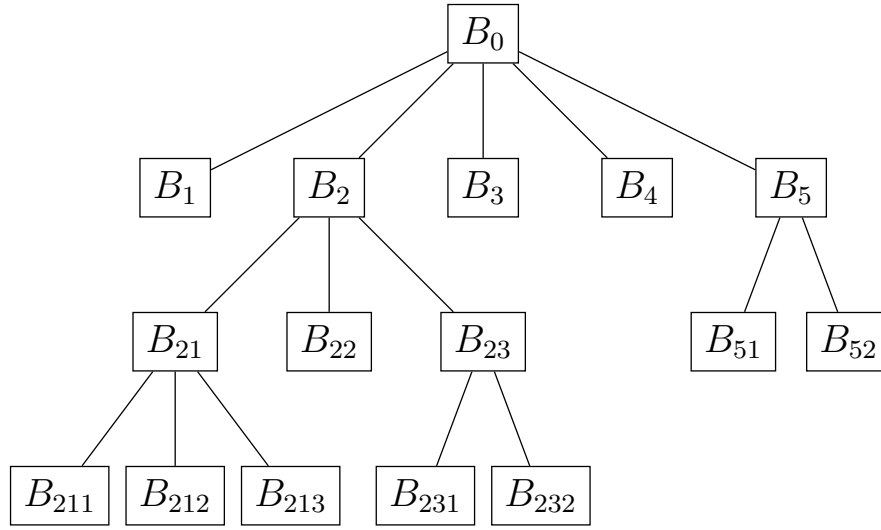
Zápis $\lfloor a \rfloor$ značí největší celé číslo menší nebo rovno a . Dolní mez z_{lb} má význam nejlepšího doposud nalezeného řešení, tj. pokud pro konkrétní rozmístění platí $z > z_{lb}$, pak $z_{lb} = z$. Horní mez z_{ub} představuje optimální řešení pro daný problém⁴. Je-li $z_{lb} = z_{ub}$, bylo nalezeno optimální řešení a algoritmus je ukončen.

Rekurzivní algoritmus

Hlavní změna oproti původní verzi spočívá v převedení metody na rekurzivní. Na každou oblast analyzovaného rozmístění se znovu aplikuje algoritmus dělení, přičemž může být rozdělena na dvě až pět oblastí. Dělení probíhá dokud nebylo dosaženo stanovené nejnižší úrovně, resp. nejvyššího počtu „zanoření“ N , nebo dokud jsou oblasti stále dostatečně velké pro umístění alespoň jedné krabice.

⁴jiná často využívaná horní mez je například Barnesova mez

Proces může být znázorněn jako stromová struktura, ve které je každá oblast představována uzlem (node). Z uzlu vychází nejvýše pět hran (branches), které reprezentují další dělení příslušné oblasti. Písmeno n značí úroveň uzlu ve stromu.



Obr. 3.7: Příklad stromové struktury.

Symetrická rozmístění

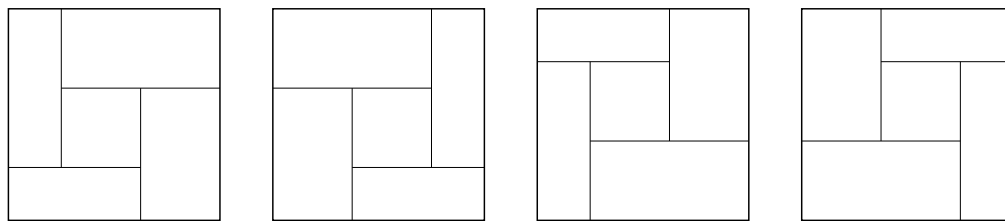
Při generování kombinací vnějších rozměrů mohou postupně vznikat symetrická rozmístění. Jsou to řešení se stejným počtem umístěných krabic, oblasti jsou však navzájem určitým způsobem rotovány či překlopeny. Obr. 3.8 zobrazuje typy symetrií, jejichž rozbořením byly stanoveny podmínky pro vnější rozměry takové, aby byla generována pouze rozmístění bez symetrických ekvivalentů. Tím došlo ke snížení doby výpočtu.

Zatímco článek [20] definuje zvlášť podmínky pro všechny typy dělení (na pět, čtyři, tři nebo dvě oblasti), [21] přichází s následující myšlenkou. Rozmístění sestávající se ze čtyř nebo tří oblastí (obr. 3.8b, 3.8c) mohou být z analýzy vyloučeny, jelikož k nim lze pomocí rekursivního algoritmu dojít sérií dělení na dvě oblasti, tzv. gilotinovými řezy. Pak je tedy možné zjednodušit podmínky tak, aby byla vytvářena pouze dělení na dvě nebo pět oblastí. V prvním případě jsou analyzována pouze rozmístění splňující

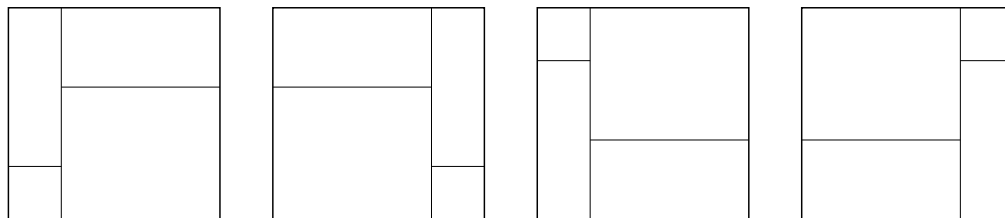
$$y_1 = y_2 = 0, \quad 0 < x_1 = x_2 = x \leq \lfloor L/2 \rfloor, \quad x \in S_L \quad (3.20)$$

při vertikálním dělení a

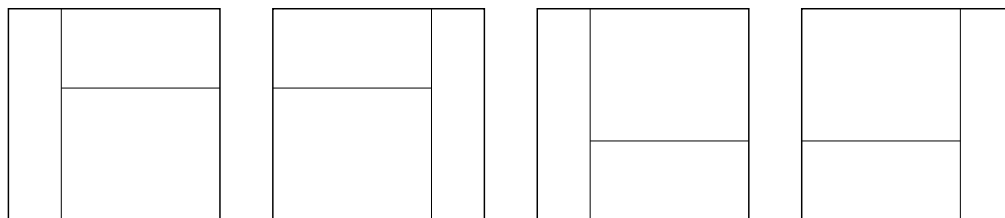
$$x_1 = x_2 = 0, \quad 0 < y_1 = y_2 = y \leq \lfloor W/2 \rfloor, \quad y \in S_W \quad (3.21)$$



(a) Symetrická rozdělení na pět oblastí.



(b) Symetrická rozdělení na čtyři oblasti.



(c) Symetrická rozdělení na tři oblasti.

Obr. 3.8: Příklady symetrických rozdělení podle [20].

při dělení horizontálním. Ve druhém případě – při dělení na pět oblastí – jsou generována pouze rozmístění splňující

$$\begin{aligned} x_1, x_2 \in S_L, \quad x_1 \leq \lfloor L/2 \rfloor, \quad x_1 < x_2, \quad x_1 + x_2 \leq L, \\ y_1, y_2 \in S_W, \quad y_1 \leq \lfloor W/2 \rfloor, \quad y_1 < y_2, \quad y_1 + y_2 \leq W. \end{aligned} \quad (3.22)$$

3.3.4 L-přístup

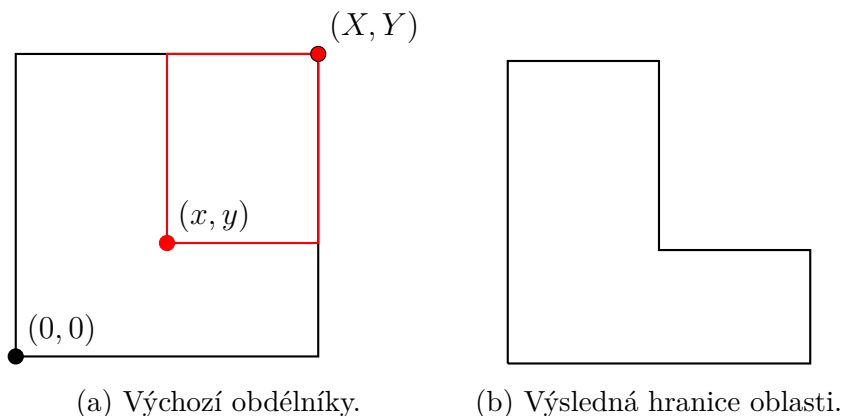
Autoři článku [22] prezentují metodu pro generování paletizačního vzoru velmi podobnou pětiblokové metodě. Hlavní myšlenkou je doplnit dělení palety na obdélníkové oblasti o možnost rozdělení na oblasti ve tvaru L⁵. Metoda je rekursivní, tedy každá vzniklá oblast se může nadále dělit na L nebo R (obdélníkové) oblasti.

L-přístupem se podařilo vyřešit případy, u kterých ostatní heuristické přístupy selhaly. Metoda rovněž poskytuje základ pro více obecný případ paletizace, a sice generování paletizačního vzoru pro různě velké krabice. Navíc ji lze použít i pro problém řezání, pokud je potřeba vyřezávat kusy ve tvaru L. Tímto problémem se do té doby zabývala málokterá dostupná literatura.

⁵odtud název L-approach

Značení oblastí

Základním prvkem, se kterým metoda pracuje, je tzv. *standardně umístěná oblast* $L(X, Y, x, y)$, kde $X \geq x$ a $Y \geq y$. Vznikne rozdílem obdélníku, jehož úhlopříčka vychází z bodu $(0, 0)$ a končí v bodě (X, Y) , a obdélníku s úhlopříčkou mezi body (x, y) a (X, Y) . Grafické znázornění je na obr. 3.9.



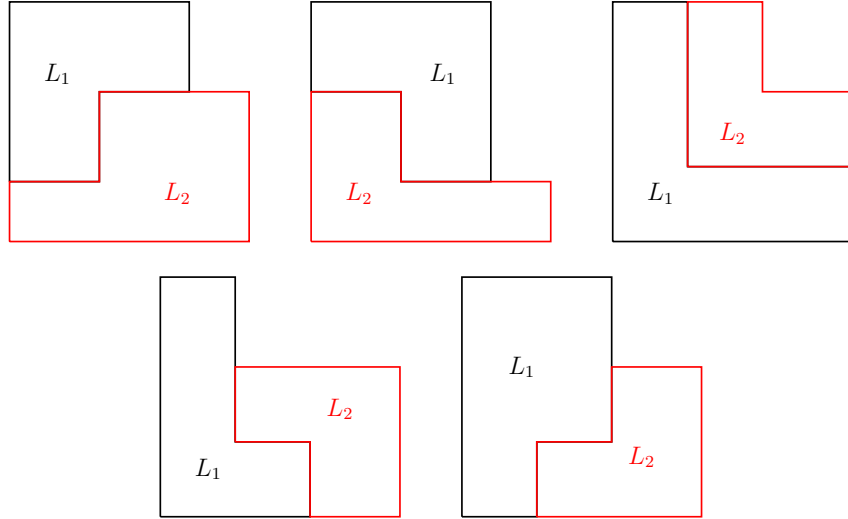
Obr. 3.9: Grafické znázornění L oblasti.

Ve speciálním případě $L(X, Y, X, Y) = R(X, Y)$ má oblast tvar obdélníku s úhlopříčkou mezi body $(0, 0)$ a (X, Y) . Obecně L oblast má tvar obdélníku pokud $x = X$ nebo $y = Y$ a nazývá se R oblast⁶.

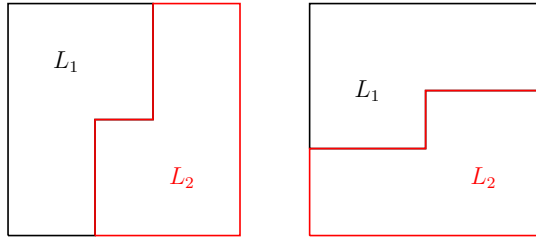
Způsoby dělení

Každá oblast může být rozdělena na dvě menší oblasti. Obr. 3.10 zobrazuje pět možných způsobů dělení L oblasti na dvě menší L oblasti. Jednotlivá rozdělení se značí jako B_k , $k = 1, \dots, 5$, ve stejném pořadí, v jakém jsou znázorněna na obrázku. Obdobně jsou na obr. 3.11 znázorněny dva způsoby dělení R oblasti na dvě menší L oblasti. Tyto se značí B_6 , B_7 .

⁶v původním článku se užívá označení degenerovaná L oblast



Obr. 3.10: Pět způsobů rozdělení L oblasti na dvě menší L oblasti.



Obr. 3.11: Dva způsoby rozdělení R oblasti na dvě menší L oblasti.

Nyní je potřeba znázorněná rozdělení vhodně matematicky popsat. Pro zadanou oblast $L = L(X, Y, x, y)$ značí P'_k , $k \in \{1, 2, 3, 4, 5\}$, množinu vnitřních parametrů rozdělení B_k . Pro $k \in \{1, 3, 5\}$ platí

$$P'_k(L) = \{p_k = (x', y') \mid 0 \leq x' \leq x, 0 \leq y' \leq y\} \quad (3.23)$$

a pro $k \in \{2, 4\}$

$$P'_2(L) = \{p_2 = (x', y') \mid 0 \leq x' \leq x, y \leq y' \leq Y\}, \quad (3.24)$$

$$P'_4(L) = \{p_4 = (x', y') \mid x \leq x' \leq X, 0 \leq y' \leq y\}. \quad (3.25)$$

Obdobně pro oblast $L = R(X, Y)$

$$P'_6(L) = \{p_6 = (x', x'', y') \mid 0 \leq x' \leq x'' \leq X, 0 \leq y' \leq Y\}, \quad (3.26)$$

$$P'_7(L) = \{p_7 = (x', y', y'') \mid 0 \leq x' \leq X, 0 \leq y' \leq y'' \leq Y\}. \quad (3.27)$$

Následně jsou pro popis každého rozdělení B_k , $k = 1, \dots, 7$, zavedeny funkce $\mathcal{L}_1(L, k, p_k)$ a $\mathcal{L}_2(L, k, p_k)$.

$$\begin{aligned}
B_1 : \mathcal{L}_1(L, 1, p_1) &= L(x, Y - y', x', Y - y) \\
\mathcal{L}_2(L, 1, p_1) &= L(X, y, X - x', y') \\
B_2 : \mathcal{L}_1(L, 2, p_2) &= L(x, Y - y, x - x', Y - y') \\
\mathcal{L}_2(L, 2, p_2) &= L(X, y', x', y) \\
B_3 : \mathcal{L}_1(L, 3, p_3) &= L(X, Y, x', y') \\
\mathcal{L}_2(L, 3, p_3) &= L(X - x', Y - y', x - x', y - y') \\
B_4 : \mathcal{L}_1(L, 4, p_4) &= L(x', Y, x, y') \\
\mathcal{L}_2(L, 4, p_4) &= L(X - x, y, X - x', y - y') \\
B_5 : \mathcal{L}_1(L, 5, p_5) &= L(x, Y, x', Y - y') \\
\mathcal{L}_2(L, 5, p_5) &= L(X - x', y, X - x, y') \\
B_6 : \mathcal{L}_1(L, 6, p_6) &= L(x'', Y, x', Y - y') \\
\mathcal{L}_2(L, 6, p_6) &= L(X - x', Y, X - x'', y') \\
B_7 : \mathcal{L}_1(L, 7, p_7) &= L(x, Y - y', x', Y - y'') \\
\mathcal{L}_2(L, 7, p_7) &= L(X, y'', X - x', y')
\end{aligned} \tag{3.28}$$

Aby byla jakákoliv oblast validní, musí mít nenulovou plochu. Proto jsou vnitřní parametry p_k omezeny na množinu $P_k(L) \subset P'_k(L)$.

$$P_k(L) = \{p_k \in P'_k(L) \mid A(\mathcal{L}_1(L, k, p_k)) > 0, A(\mathcal{L}_2(L, k, p_k)) > 0\}, \tag{3.29}$$

kde $k \in 1, 2, \dots, 7$ a $A(L)$ je plocha oblasti L .

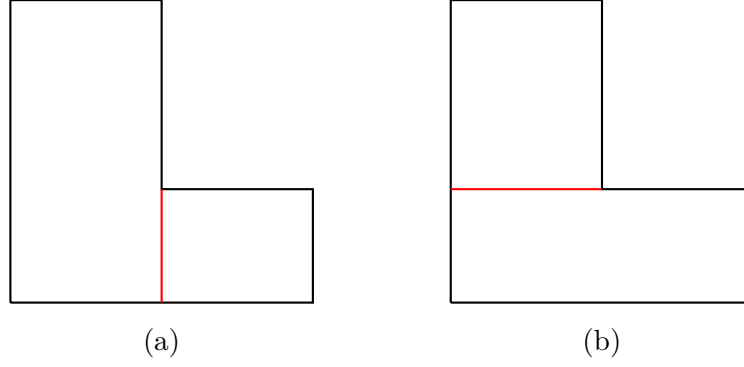
Dolní a horní mez

Stejně jako u pětiblokové metody i zde je zavedena dolní mez z_{lb} a horní mez z_{ub} . V případě R oblasti zůstává dolní mez z_{lb} stejná jako u pětiblokové metody.

$$z_{lb}(L) = \max\{\lfloor X/l \rfloor \lfloor Y/w \rfloor, \lfloor X/w \rfloor \lfloor Y/l \rfloor\} \tag{3.30}$$

Změna nastává u L oblasti. Tato se rozloží na dvě R oblasti, které se vyhodnocují zvlášť.

$$\begin{aligned}
z_{lb}(L) &= \max\{z_{lb}(R(x, Y)) + z_{lb}(R(X - x, y)), \\
&\quad z_{lb}(R(x, Y - y)) + z_{lb}(R(X, y))\}
\end{aligned} \tag{3.31}$$



Obr. 3.12: Způsoby dělení L oblasti na dvě R oblasti.

Omezení množiny rozdělení

Výpočetní náročnost algoritmu je snížena omezením parametrů X, Y, x a y na množiny S_L, S_W (případně R_L, R_W). Dále jsou pro tyto parametry zavedena následující pravidla, která zamezí generování ekvivalentních (symetrických) rozdělení.

1. $X \geq x$ a $Y \geq y$
2. $X \geq Y$ a $x \geq y$
3. $X = x$ pouze tehdy, když $Y = y$

Vzhledem k těmto omezením je nutno upravit definice rozdělení B_k , $k = 1, \dots, 7$, v rov. 3.28. Rozdíly parametrů (např. $X - x$) ve funkcích $\mathcal{L}_1(L, k, p_k)$ a $\mathcal{L}_2(L, k, p_k)$ nemusí ležet v množinách, na které jsou tyto parametry omezeny (S_L a S_W). Rozdíl je tedy nahrazen operátorem \ominus .

$$A \ominus a = \max\{r \mid r \leq A - a, r \in R\}, \quad (3.32)$$

kde R je množina, na kterou je příslušný parametr omezen. Tedy $r \in R = S_L$ pro $X \ominus x, X \ominus x', X \ominus x'', x \ominus x'$ a $r \in R = S_W$ pro $Y \ominus y, Y \ominus y', Y \ominus y'', y \ominus y'$.

4 Implementace knihovny pro generování paletizačních vzorů

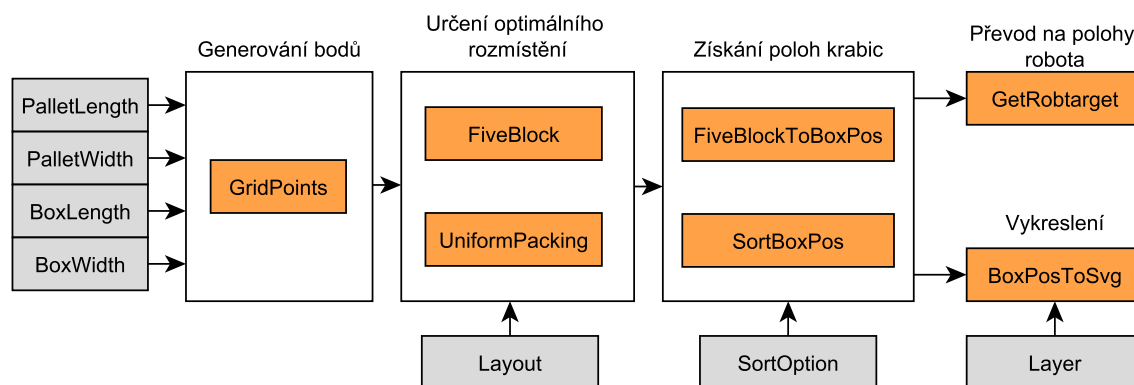
Tato kapitola popisuje způsob implementace knihovny pro generování paletizačních vzorů. Knihovna byla vytvořena ve vývojovém prostředí Automation Studio 4.6. Prostředí obsahuje nástroje potřebné k vývoji softwaru, umožňuje pracovat s hardwarem společnosti B&R Automation a zároveň umožňuje hardware simulovat.

4.1 Koncepce

Celý proces generování paletizačních vzorů je rozdělen do čtyř částí:

1. generování normalizovaných bodů,
2. určení optimálního rozmístění,
3. získání poloh umístěných krabic,
4. výpočet koncových poloh robota.

Každá část je realizována samostatnou funkcí. Funkce se volají z funkčního bloku RTPalletBasic, čímž je zajištěna správná posloupnost operací se správnými parametry. Vstupy pro návrh paletizačního vzoru jsou rozměry palety a krabice, metoda návrhu a způsob řazení krabic. Výstupem je seznam koncových poloh robota.



Obr. 4.1: Koncepce knihovny RTPallet

4.2 Názvosloví

Tvorba knihovny podléhá vnitřním normám společnosti B&R Automatizace, která stanoví i způsob pojmenování funkcí, proměnných, konstant apod.

Název knihovny se skládá z předpony (prefixu), který určuje do jaké kategorie knihovna spadá, a vlastního názvu. V tomto případě je knihovna nazvána RTPallet, kde RT představuje označení projektu ROBOTemplate.

Typové definice, názvy funkcí a funkčních bloků začínají názvem knihovny. Poté následuje vlastní název. Celková délka je omezena na 32 znaků.

Názvy konstant a výčtů rovněž vyplývají z názvu knihovny a v konkrétním případě mají podobu „rtPALLET_“, přičemž každé další slovo je psáno velkými písmeny a odděleno podtržítkem. Délka je také omezena na 32 znaků.

4.3 Datový typ pro rozměry

Rozměry palet jsou normovány. Tab. 4.1 uvádí rozměry šesti typů palet podle normy ISO 6780:2003. Průzkumem sortimentu prodejců s obalovým materiálem bylo zjištěno, že jednotlivé rozměry kartónových krabic se pohybují mezi 150 a 600 mm. Na základě těchto poznatků byl stanoven datový typ pro veškeré proměnné nesoucí informaci o rozměru či poloze na celočíselný unsigned int o délce 2 B. Hodnoty jsou udávány v milimetrech, což je v souladu s běžnou technickou praxí.

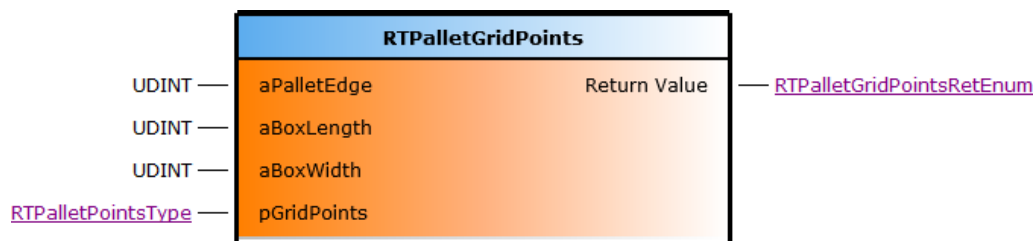
Tab. 4.1: Rozměry palet podle ISO 6780:2003. Převzato z [24].

Rozměry [mm]	Nejčastější výskyt
1016×1219	Severní Amerika
1000×1200	Evropa, Asie
1165×1165	Austrálie
1067×1067	Severní Amerika, Evropa, Asie
1100×1100	Asie
800×1200	Evropa (EUR2)

4.4 Normalizované body

Vytvoření bodů podle rov. 3.1 zajišťuje funkce RTPalletGridPoints. Obsahuje čtyři parametry: jeden z rozměrů palety PalletEdge, délku BoxLength a šířku BoxWidth krabice a ukazatel GridPoints na strukturu, do které mají být uloženy vytvořené body. Návrátová hodnota funkce je výčtového typu, slouží jako indikace případné chyby při výpočtu.

Parametr PalletEdge představuje horní mez, do které se mohou body generovat. Struktura RTPalletPointsType obsahuje počet vygenerovaných bodů Count a pole s hodnotami Points. Délka tohoto pole je statická, jelikož v rámci B&R Automatizace není možné použít funkce pro dynamické přidělení paměti. V závislosti na zadaných parametrech se však počet vygenerovaných bodů liší a není předem znám.



Obr. 4.2: Funkce RTPalletGridPoints

Délka pole hodnot byla stanovena s uvažováním „nejhoršího možného případu“, tedy $\text{PalletEdge} = 1200^1$, $\text{BoxLength} = 1$, $\text{BoxWidth} = 1$. Byť je takové zadání v praxi nepravděpodobné, poskytuje určitou představu o počtu generovaných bodů. S uvedeným zadáním by funkce vygenerovala 1201 bod. Délka pole byla tedy stanovena na 2000 prvků. V případě, že by nebyla dostatečná, je možné ji upravit novou kompilací knihovny.

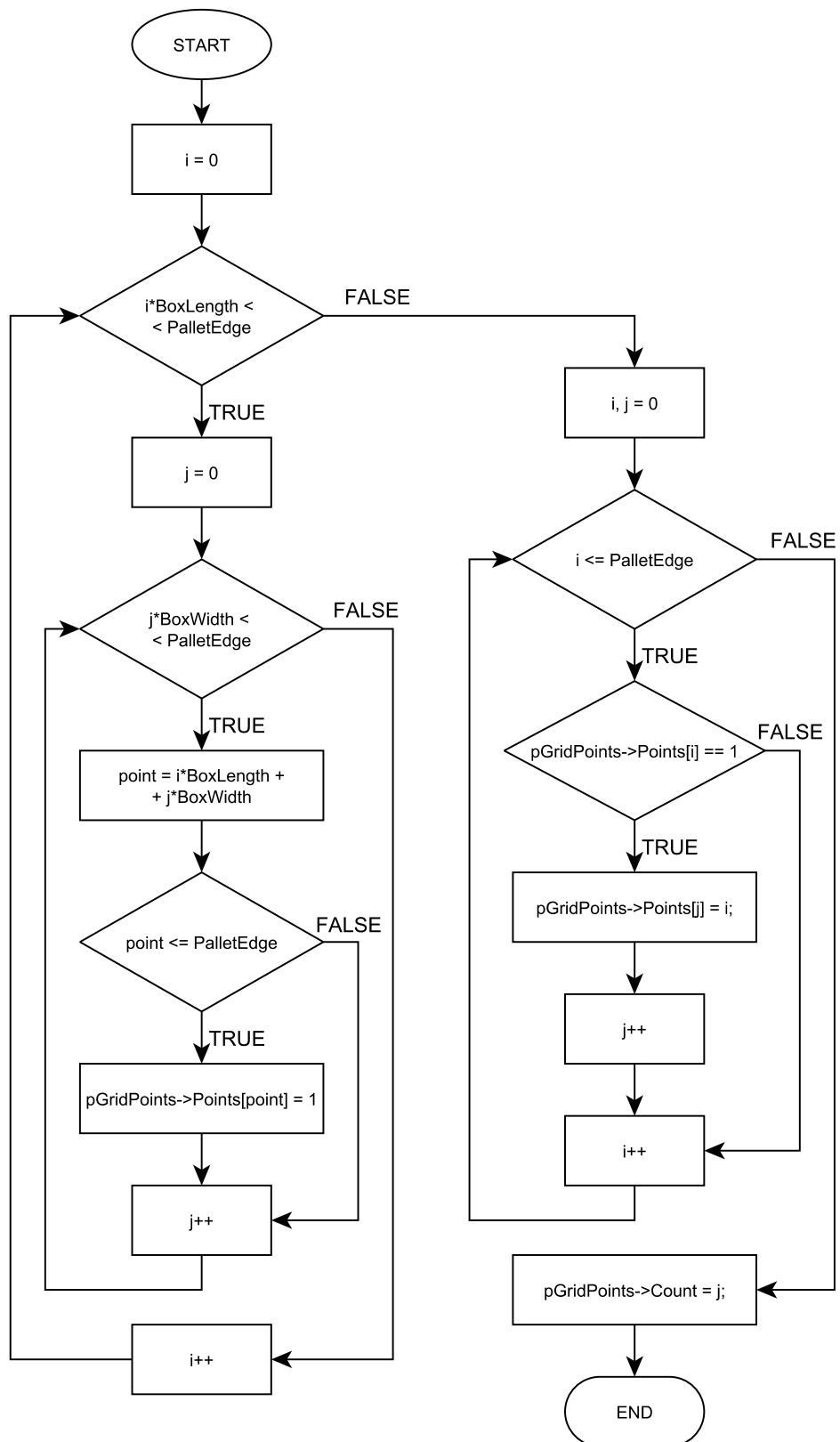
Činnost funkce popisuje vývojový diagram na obr. 4.3. Pro přehlednost diagram nezahrnuje kontroly vstupních parametrů, ošetření mezí atd.

Nejdříve je pole $\text{pGridPoints} \rightarrow \text{Points}$ vynulováno. Dále jsou pomocí dvou cyklů postupně generovány body podle rov. 3.1. Vygenerované hodnoty v tuto chvíli slouží jako index prvku v poli. Příslušnému prvku je nastavena hodnota 1, která značí, že daný bod do posloupnosti patří.

Další fáze přeskládá prvky v poli tak, aby hodnoty patřící do posloupnosti následovaly bezprostředně za sebou. Zároveň ve výsledném poli již hodnota bodu není představována indexem prvku, nýbrž je uložena přímo v hodnotě prvku.

Tento přístup efektivně řeší duplicitní body, které jistě vznikají „pouhým“ generováním hodnot pomocí dvou cyklů a jejich zapisováním do pole. Například při zadání $\text{PalletEdge} = 20$, $\text{BoxLength} = 4$, $\text{BoxWidth} = 3$ by hodnota 12 byla poprvé generována jako $0 \cdot 4 + 4 \cdot 3$ a podruhé jako $3 \cdot 4 + 0 \cdot 3$.

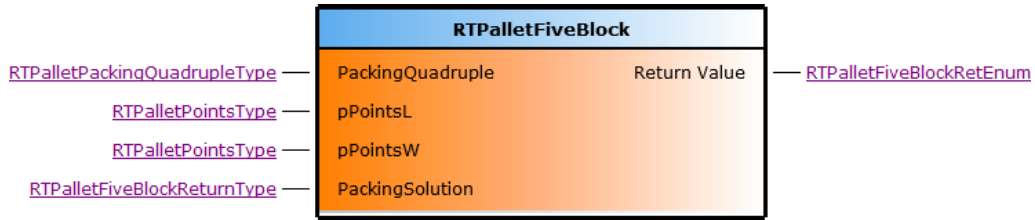
¹rozměr vychází ze standardní europalety 1200×800 mm



Obr. 4.3: Vývojový diagram funkce RTPalletGridPoints

4.5 Určení optimálního rozmístění

Návrh optimálního rozdělení palety na pět oblastí realizuje funkce RTPalletFiveBlock. Činnost vychází z metody popsané v kap. 3.3.2. Rozhraní funkce je zobrazeno na obr. 4.4. Parametr PackingQuadruple obsahuje rozměry PalletL, PalletW palety a BoxL, BoxW krabice. Struktury PointsL a PointsW vygenerované body pomocí funkce RTPalletGridPoints a struktura PackingSolution slouží pro zapsání výsledku. Prvky datového typu RTPalletFiveBlockReturnTypu uvádí tab. 4.2.



Obr. 4.4: Funkce RTPalletFiveBlock

Po zavolání funkce nejdříve proběhne kontrola zadaných ukazatelů. Rozměry nesmí být nulové a musí platit $PalletL \geq PalletW \geq BoxL \geq BoxW$. Rovněž struktury bodů nesmí být prázdné. Při chybně zadaných parametrech je funkce ukončena s návratovou hodnotou, která blíže specifikuje důvod ukončení.

Je-li kontrola parametrů úspěšná, dojde k vynulování struktury PackingSolution a je určen počet umístěných krabic v případě tzv. homogenního rozložení². To je takové, kdy krabice jsou rozloženy jedna vedle druhé se stejnou orientací po celé ploše palety. Porovnávají se dvě možné orientace krabic: první, kdy je delší hrana krabic rovnoběžná s delší hranou palety, a druhá, kdy jsou krabice otočeny o 90°. Řešení s větším počtem umístěných krabic je uloženo jako dolní mez z_{lb} .

Dále je vypočtena horní mez z_{ub} podle rov. 3.18. Následuje procházení možných řešení s pěti oblastmi. Jednotlivým oblastem jsou nastavovány velikosti z množin PointsL, PointsW, kombinace s překrývajícími se oblastmi jsou vynechány a u ostatních – vyhovujících – kombinací je vyhodnocen počet umístěných krabic při uvažování jejich orientace podle obr. 3.5. Pro nejlepší řešení funkce vrátí: celkový počet umístěných krabic BoxCount a počet krabic v jednotlivých oblastech, orientaci krabic ve středové oblasti, jelikož tato není pevně určena, rozměry oblastí. Jestliže bylo nejlepšího výsledku dosaženo homogenním rozložením, je nastaven příznak Homogeneous.

Pokud nějaké řešení obsahuje počet umístěných krabic větší nebo roven horní mezi z_{ub} , je nastaven příznak IsOptimal a funkce ukončena. V opačném případě dojde k ukončení až po vyhodnocení všech kombinací.

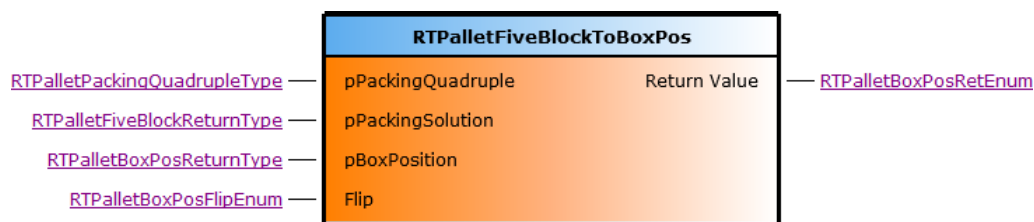
²homogeneous packing

Tab. 4.2: Prvky datového typu RTPalletFiveBlockReturnType

Proměnná	Datový typ	Popis
BoxCount	UDINT	celkový počet umístěných krabic
BoxCountPerBlock	UDINT[4]	počet krabic v jednotlivých oblastech
BlockDim	RTPalletFiveBlockOuterDimType	rozměry jednotlivých oblastí
Orientation	BOOL	orientace krabic ve 3. oblasti, nebo v případě homogenního řešení orientace všech boxů
IsOptimal	BOOL	nalezené řešení je optimální podle použitého kritéria
Homogeneous	BOOL	nalezené řešení je homogenní

4.6 Určení pozice krabic

Funkce RTPalletFiveBlock pro návrh optimálního rozmístění vrací velikosti jednotlivých oblastí a počet umístěných krabic. Pro další práci – například vytvoření procedury pro robota – je však nutné znát polohu každé krabice. K tomu slouží funkce RTPalletFiveBlockToBoxPos. Mezi její parametry patří PackingQuadruple obsahující rozměry palety a krabice, struktura PackingSolution s velikostmi oblastí a struktura BoxPosition pro uložení výsledku.



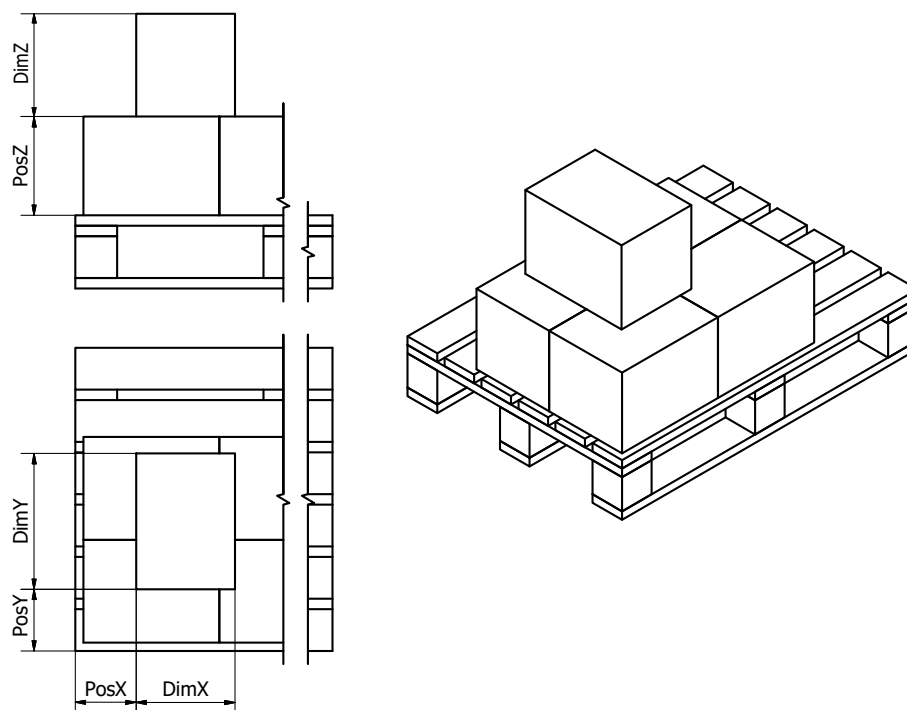
Obr. 4.5: Funkce RTPalletFiveBlockToBoxPos

Struktura BoxPosition obsahuje proměnnou BoxCount a pole BoxPos. Proměnná BoxCount značí počet uložených pozic, resp. počet využitých prvků v poli BoxPos. Samotné pole pro uložení pozic má, stejně jako pole pro uložení generovaných bodů u funkce RTPalletGridPoints, statickou délku. Jeho datový typ je ukázán v tab. 4.3. Skládá se z proměnných PosX, PosY, PosZ, které odpovídají pozici levého spodního rohu krabice vzhledem k levému spodnímu rohu palety, a dále rozměrů DimX, DimY, DimZ, které prakticky uchovávají informaci o orientaci krabice. Obr. 4.6 vysvětluje význam proměnných graficky.

Tab. 4.3: Prvky datového typu RTPalletBoxPosType

Proměnná	Datový typ	Popis
PosX	UDINT	pozice krabice vzhledem k levému okraji palety
PosY	UDINT	pozice krabice vzhledem ke spodnímu okraji palety
PosZ	UDINT	pozice krabice nad paletou
DimX	UDINT	délka krabice
DimY	UDINT	šířka krabice
DimZ	UDINT	výška krabice

Poslední parametr Flip umožňuje horizontálně převrátit navržené rozmístění. Důvod pro zavedení této operace je prostý. Krabice jsou na paletu skládány zpravidla ve více vrstvách. Pokud budou jednotlivé vrstvy navrženy stejným algoritmem, bude jejich rozmístění stejné – krabice budou položeny přesně na sobě bez překrytí. Z hlediska stability je však žádoucí, aby mezi vrstvami k překrytí docházelo, čehož lze konkrétně pro rozmístění s pěti oblastmi jednoduše docílit právě horizontálním překlopením.

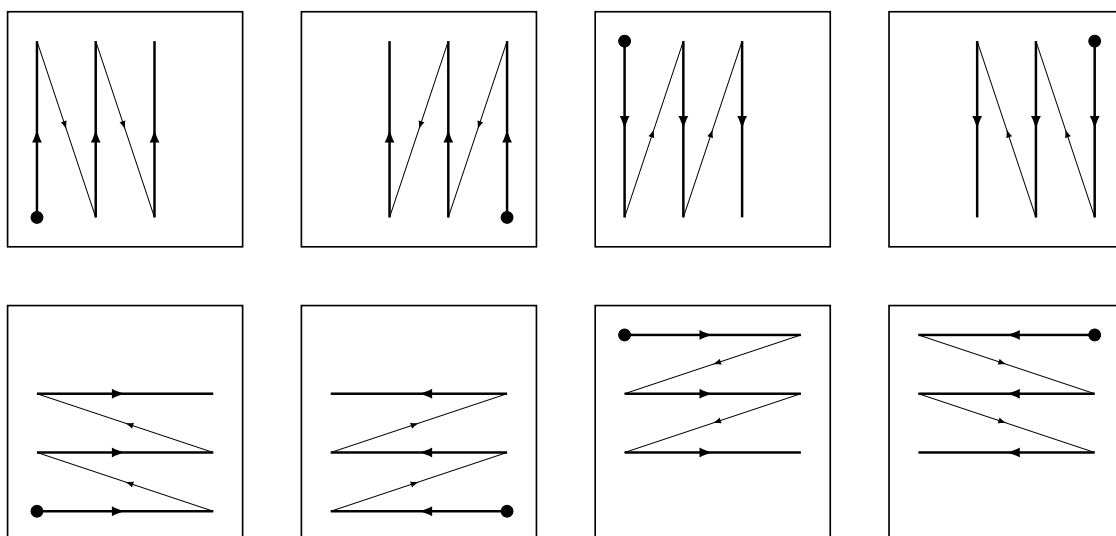


Obr. 4.6: Význam proměnných v datovém typu RTPalletBoxPosType.

4.7 Seřazení krabic

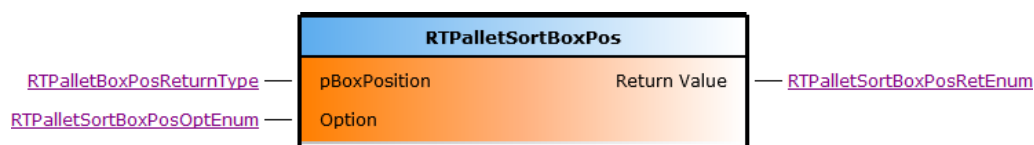
Funkce pro návrh rozmístění krabic na paletě nijak nezohledňují pořadí, v jakém mají být krabice skládány. Jak již bylo popsáno, navržené rozmístění je uloženo v podobě pozice každé krabice. Má-li být vytvořena pohybová procedura pro robota, je potřeba nejdříve seřadit seznam krabic tak, aby při nakládání na paletu nedocházelo ke kolizím.

Příklady vhodných pořadí nakládání jsou zobrazeny na obr. 4.7. Všechny naznačené možnosti začínají v některém rohu palety, krabice zaplňují paletu vždy jen v jednom směru.



Obr. 4.7: Možná pořadí nakládání krabic na paletu v rámci jedné vrstvy.

Pro seřazení seznamu krabic byla vytvořena funkce `RTPalletSortBoxPos`. Vstupními parametry jsou struktura `pBoxPosition` a výčtový typ `Option`, kterým se volí způsob řazení. Funkce provádí úpravy přímo nad zadaným seznamem. Návrátová hodnota slouží nese informace o průběhu (úspěch, špatně zadané vstupní parametry).



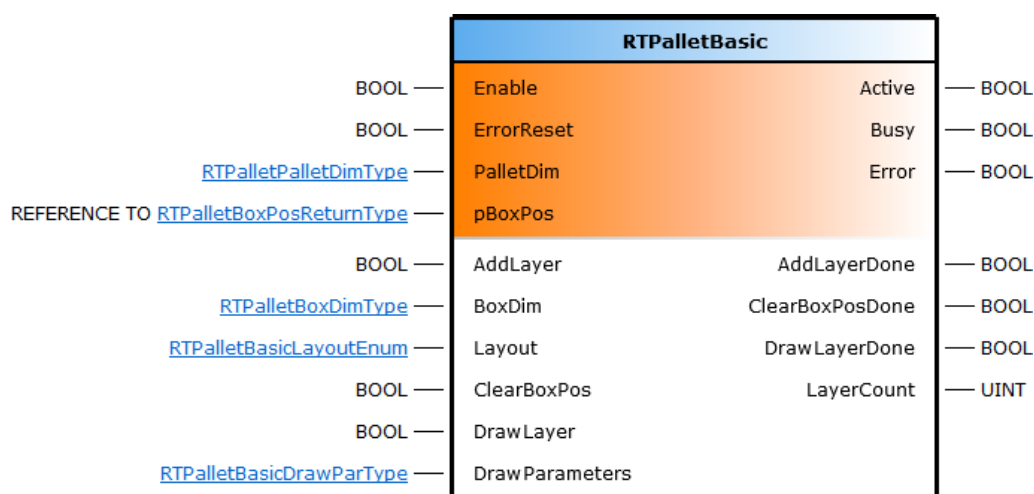
Obr. 4.8: Funkce `RTPalletSortBoxPos`.

Řazení seznamu je realizováno standardní funkcí `qsort` z knihovny `stdlib.h`. Jejími parametry jsou ukazatel na pole, které má být seřazeno, počet prvků v poli, délka (velikost) jednoho prvku a ukazatel na tzv. srovnávací funkci, která porovná dva

prvky pole. V závislosti na parametru Option se funkce qsort zavolá vždy s jinou srovnávací funkcí. V souladu s obr. 4.7 je dostupných osm způsobů řazení. Výsledný seznam je seřazen vzestupně nejdříve podle vrstev a následně podle zvoleného způsobu.

4.8 Funkční blok RTPalletBasic

Funkční blok RTPalletBasic zjednodušuje použití knihovny RTPallet. Zajišťuje volání funkcí popsanych v předchozích kapitolách se správnými parametry a uchovávání výsledků.



Obr. 4.9: Funkční blok RTPalletBasic.

Po zapnutí funkčního bloku vstupem Enable jsou zkontrolovány parametry PalletDim a pBoxPos. Rozměry palety musí být nenulové a musí platit $Length > Width$. Struktura RTPalletBoxPosReturnType, na kterou ukazuje parametr pBoxPos slouží pro uchování výsledku – pozic jednotlivých krabic (struktura bude nadále nazývána jako „výstupní struktura“). Pokud jsou parametry zadány správně, blok je připraven k použití, což je signalizováno hodnotou TRUE na výstupu Active.

S blokem lze pracovat prostřednictvím několika vstupů. Vstup AddLayer slouží pro přidání vrstvy krabic na paletu. K příkazu se váží parametry BoxDim, který obsahuje rozměr krabice a platí pro něj stejné podmínky jako pro rozměry palety, a Layout, který umožňuje volit metodu použitou pro návrh rozmístění krabic. Jsou dostupné tři možnosti:

1. rozdělení na pět oblastí,
2. homogenní rozmístění, krabice orientovány delší hranou k delší hraně palety,
3. homogenní rozmístění, krabice orientovány kratší hranou k delší hraně palety.

Pokud je zvolena první možnost, jsou postupně zavolány funkce RTPalletFiveBlock a RTPalletFiveBlockToBoxPos. Pro druhou a třetí možnost je zavolána funkce RTPalletUniformPacking. Výsledkem jsou pozice krabic ve výstupní struktuře. Nová vrstva je vytvořena vždy nad poslední vrstvou. Počet vytvořených vrstev udává výstup LayerCount. Dokončení příkazu AddLayer je signalizováno hodnotou TRUE na výstupu AddLayerDone.

Další vstup ClearBoxPos umožňuje všechny doposud vytvořené vrstvy. Po aktivaci vstupu je vynulován počet vrstev LayerCount, vymazán obsah výstupní struktury a rovněž vymazán obsah proměnné, na kterou ukazuje parametr *DrawParameters.pString*.

Vstup DrawLayer umožňuje vytvořit popis vybrané vrstvy ve formátu SVG. Vytvořený řetězec je možné využít pro vykreslení dané vrstvy. K příkazu se váží parametry ve struktuře DrawParameters. Jejich přehled uvádí tab. 4.4.

Tab. 4.4: Parametry vstupu DrawLayer

Název	Typ	Popis
Layer	USINT	vrstva, která má být vykreslena
pString	UDINT	ukazatel na pole, kde má být uložen výsledek
StringLen	UDINT	délka pole pro uložení výsledku
Normalize	BOOL	umožňuje přepočítat všechny rozměry tak, aby výsledný obraz zaplnil oblast Length×Width
Length	UINT	délka oblasti pro přepočet rozměrů
Width	UINT	šířka oblasti pro přepočet rozměrů

Výsledný SVG řetězec se skládá z prvků **rect** pro vykreslení obdélníků představující krabice a prvků **text** pro vykreslení pořadí nakládání jednotlivých krabic.

4.9 Převod poloh krabic na polohy robota

Polohu robota lze v jazyce RAPID definovat dvěma způsoby, a to buď zadáním úhlů natočení os robota (jointtarget), nebo zadáním souřadnic cílového bodu vzhledem k referenčnímu souřadnému systému (robtargt). Druhý způsob je pro danou úlohu – umístění krabic na paletu – výhodnější, jelikož ve struktuře RTPalletBoxPosType, která je zdrojem pro určení koncových poloh robota, jsou známy souřadnice krabic vzhledem k levému spodnímu rohu palety.

Datový typ robtarget se sestává ze čtyř částí:

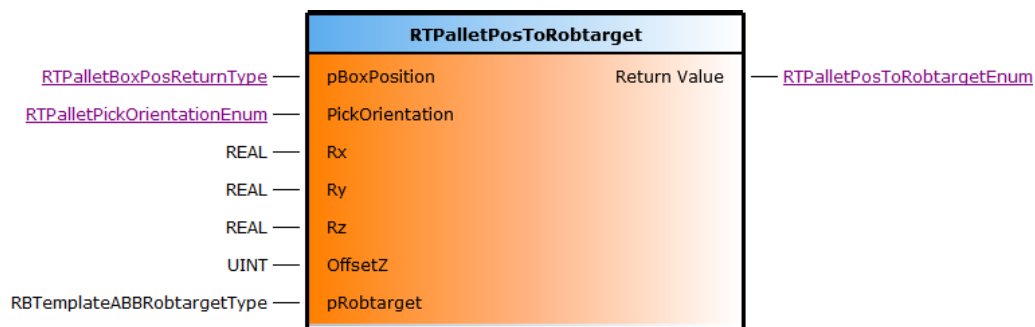
- trans – souřadnice X, Y, Z nástroje v referenčním souřadném systému,
- rot – orientace nástroje vzhledem k referenčnímu souřadnému systému,

- robconf – konfigurace os robota,
- extax – poloha externích os.

Souřadnice (x, y, z) nástroje jsou vyjádřeny v milimetrech. Orientace nástroje je vyjádřena kvaterniony (q_1, q_2, q_3, q_4) .

Některé cílové polohy jsou dosažitelné více způsoby natočení os robota. Konfigurace robconf jednoznačně určuje, který způsob natočení má být použit. Obsahuje čtyři složky (cf_1, cf_4, cf_6, cf_x) . Jejich hodnoty představují číslo kvadrantu a vymezují rozsah natočení příslušné osy, například pro 6osého jsou kvadranty definovány po 90° , tedy pro kvadrant 0 je rozsah natočení $0-90^\circ$, kvadrant 1 rozsah $90-180^\circ$ atd.

Pozice krabic je potřeba převést na cílové polohy robota. K tomu slouží funkce RTPalletPosToRobtarget. Vstupem funkce je ukazatel pBoxPosition na seznam pozic, které mají být převedeny. Převedené pozice jsou ukládány do struktury typu RBTemplateABBRobotType, která je součástí knihovny ROBOTemplate. Struktura obsahuje prvky odpovídající datovému typu robtarget jazyka RAPID.



Obr. 4.10: Funkce RTPalletPosToRobtarget.

Souřadnice (x, y, z) jsou z prvků struktury RTPalletBoxPosType (tab. 4.3) vypočteny takto

$$x = PosX + DimX/2, \quad (4.1)$$

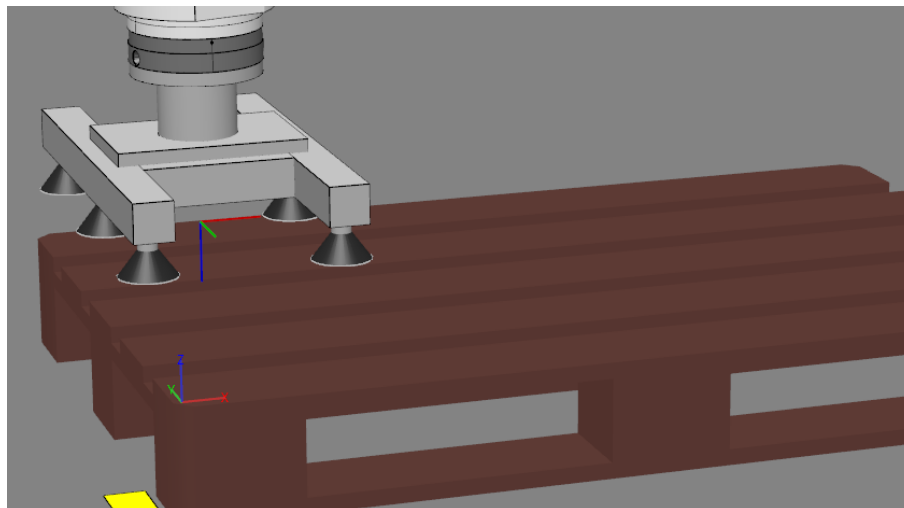
$$y = PosY + DimY/2, \quad (4.2)$$

$$z = PosZ + DimZ + OffsetZ, \quad (4.3)$$

což odpovídá středu požadované pozice ve směru os X, Y a výšce poslední vrstvy zvětšené o výšku krabice ve směru osy Z. Parametr OffsetZ umožňuje tuto výšku upravit například o toleranci rozměrů krabice či kompenzovat rozdíl mezi horní plochou krabice a středem nástroje.

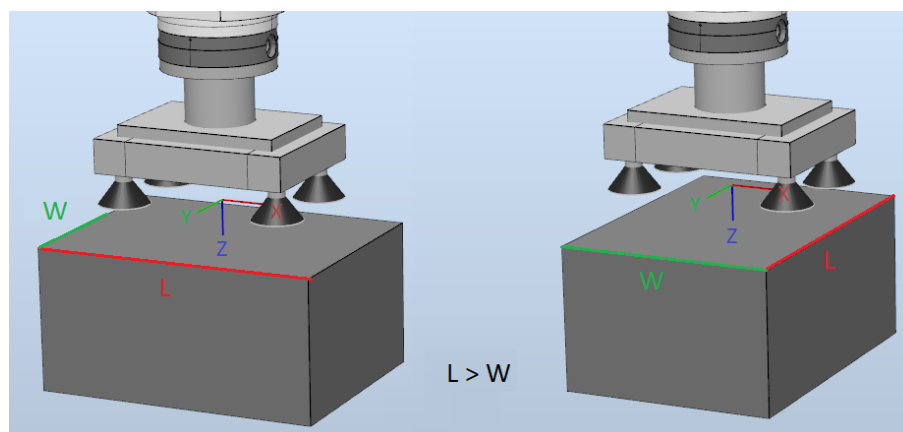
Aby bylo možné správně určit orientaci nástroje, je funkce doplněna o parametry Rx, Ry a Rz představující natočení souřadného systému nástroje vzhledem k referenčnímu souřadnému systému. Natočení je vyjádřeno ve stupních. Příklad uvádí obr. 4.11. Referenční souřadný systém je umístěn v levém spodním rohu palety

s naznačenou orientací. Zvolená orientace nástroje vyhovuje potřebám nakládání na paletu (všechny osy nástroje jsou rovnoběžné s příslušnými osami referenčního souřadného systému). Rotace nástroje kolem jednotlivých os vzhledem k referenčnímu souřadnému systému jsou $R_x = 180^\circ$, $R_y = 0^\circ$, $R_z = 0^\circ$.



Obr. 4.11: Příklad natočení souřadného systému nástroje.

Parametr PickOrientation definuje způsob, jakým jsou krabice uchopeny, a je nezbytný pro správný výpočet orientace nástroje. Parametr může nabývat dvou hodnot. Krabice je uchopena delší hranou podél osy X nástroje nebo delší hranou podél osy Y nástroje.



Obr. 4.12: Dva způsoby uchopení krabice.

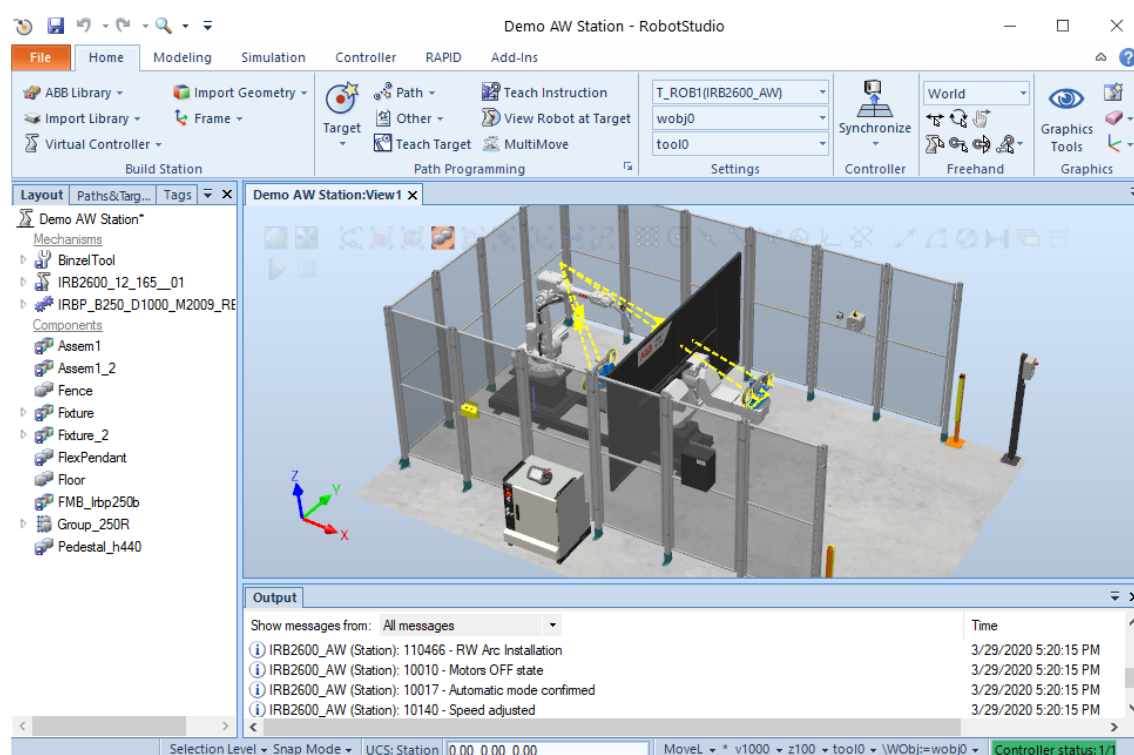
5 Digitální dvojče

Digitální dvojče je přesným virtuálním modelem stroje nebo výrobního zařízení [25]. Tato kapitola popisuje tvorbu virtuálního paletizačního pracoviště v prostředí RobotStudio a jeho propojení s PLC.

5.1 RobotStudio

RobotStudio je software pro simulaci a programování robotů od společnosti ABB. Je založeno na virtuálním řídicím systému ABB Virtual Controller – přesné kopii opravdového softwaru, který řídí roboty ve výrobě. Umožňuje tak realistické simulace s využitím programů a konfiguračních souborů identických s těmi, které jsou využívány ve výrobě [26].

Software umožňuje import libovolného 3D modelu z běžných CAD formátů, například STEP či IGES. Objektům mohou být přiřazeny fyzikální vlastnosti. Kromě toho je možné vytvářet tzv. inteligentní komponenty¹, které v rámci simulace umožňují provádět aritmetické a logické operace, snímání, pohyby objektů či parametrické modelování. Lze tak vytvářet virtuální kopie komplexních robotických pracovišť.



Obr. 5.1: Ukázka prostředí RobotStudio.

¹smart components

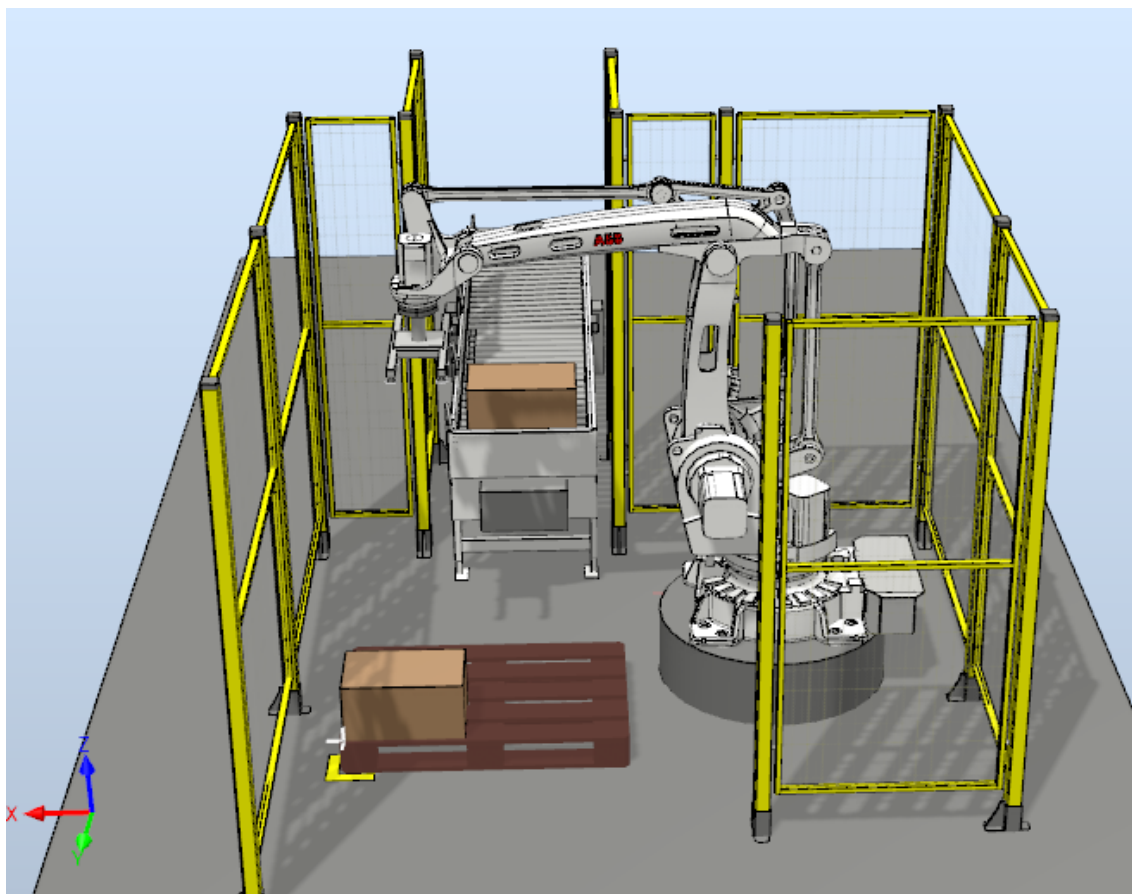
5.2 Tvorba scény

Rozvržení virtuální paletizační buňky je značně inspirováno paletizačními buňkami prezentovanými v kap. 1. Scéna však bude sloužit pouze pro ověření funkčnosti a představení knihovny pro generování paletizačních vzorů.

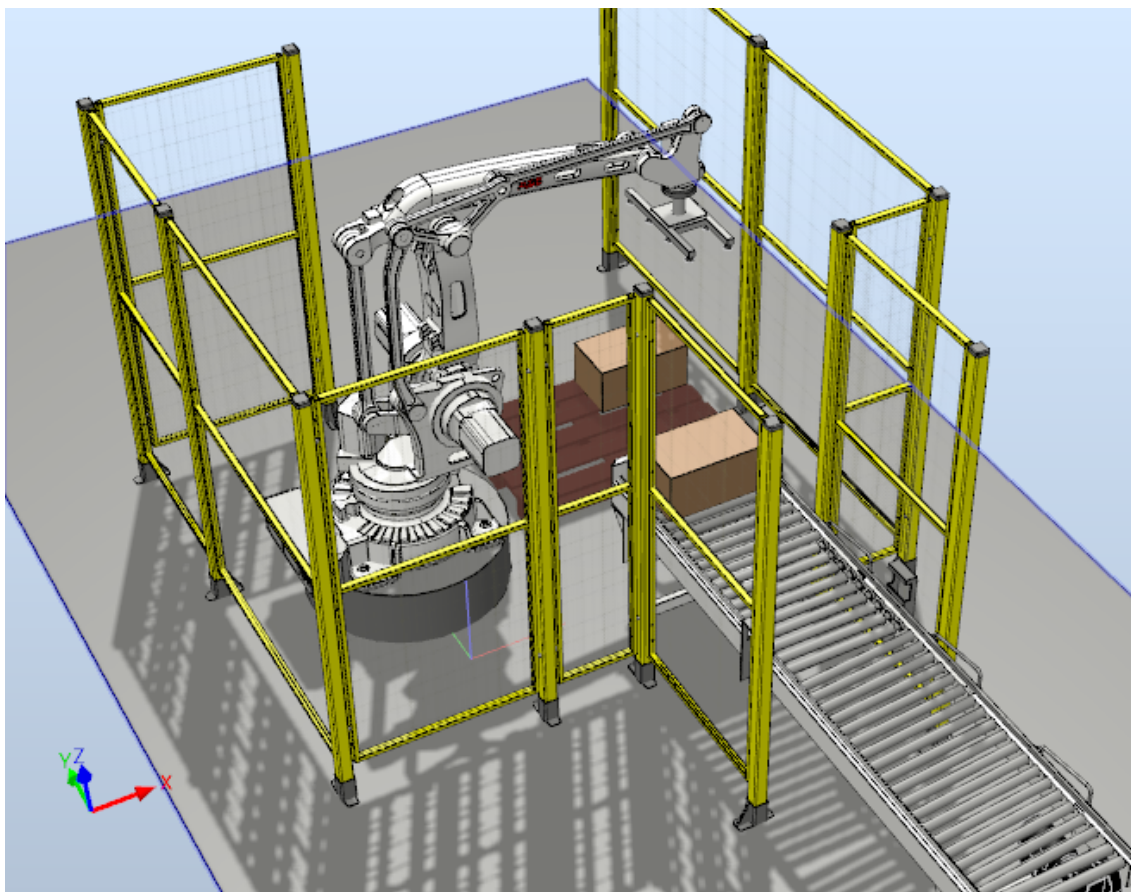
Z portfolia robotů ABB byl zvolen kompaktní průmyslový robot IRB 460 o nosnosti 110 kg a dosahu 2,4 m. Tento robot je svou 4osou konstrukcí určen právě pro paletizování a manipulaci s materiálem. Model robota je součástí standardní knihovny RobotStudio.

Buňka nepočítá s automatickou výměnou palet. Předpokládá se manuální obsluha. Před započítím paletizace je prázdná paleta umístěna na označené místo, naplněná paleta je odvezena např. vysokozdvizným vozíkem.

Scéna je doplněna o oplocení, což jí dodává na reálném vzhledu. Model oplocení je možné nalézt v uživatelské knihovně RobotStudio (Add-Ins–RobotApps). Výslednou podobu paletizační buňky zobrazují obr. 5.2 a 5.3.



Obr. 5.2: Výsledná podoba virtuálního pracoviště, pohled 1.

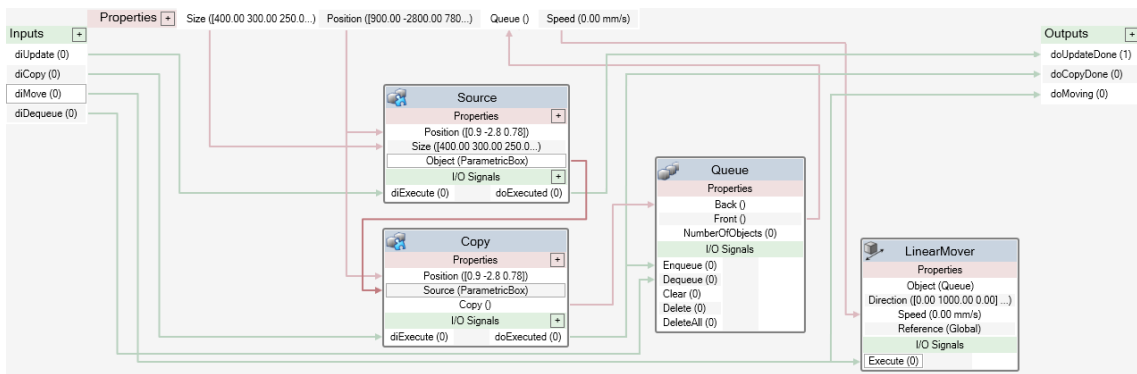


Obr. 5.3: Výsledná podoba virtuálního pracoviště, pohled 2.

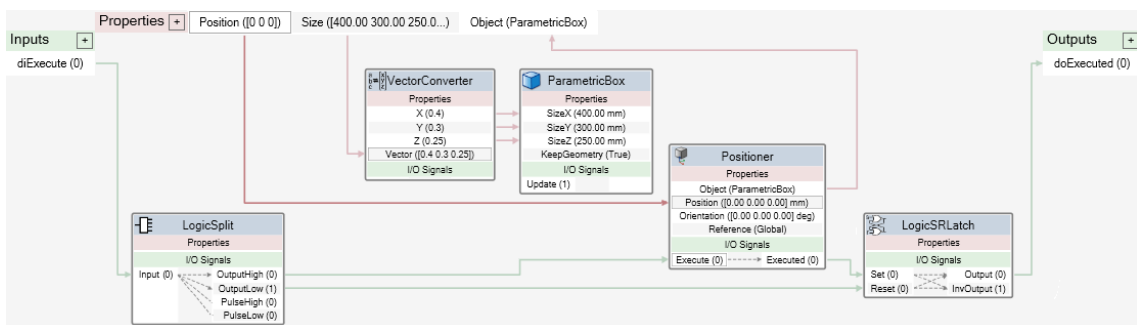
5.3 Generování krabic

Krabice jsou generovány dynamicky pomocí vytvořené inteligentní komponenty Box. Tu lze rozdělit na dvě hlavní části. První umožňuje změnu velikosti a polohy zdrojového objektu krabice podle zadaných parametrů. K tomu jsou využity prvky z nabídky RobotStudia ParametricBox a Positioner. Zdrojový objekt je skrytý. Druhá část umožňuje pomocí prvku Source ze zdrojového objektu vytvořit kopii a zobrazit ji pomocí prvku Show. Vytvořené kopie jsou uchovány ve frontě Queue. Fronta zachovává pořadí, v jakém byly kopie vytvářeny, a rovněž umožňuje se všemi kopiemi pracovat jako s jedním objektem. Těchto vlastností je využito v následujících částech virtuálního pracoviště.

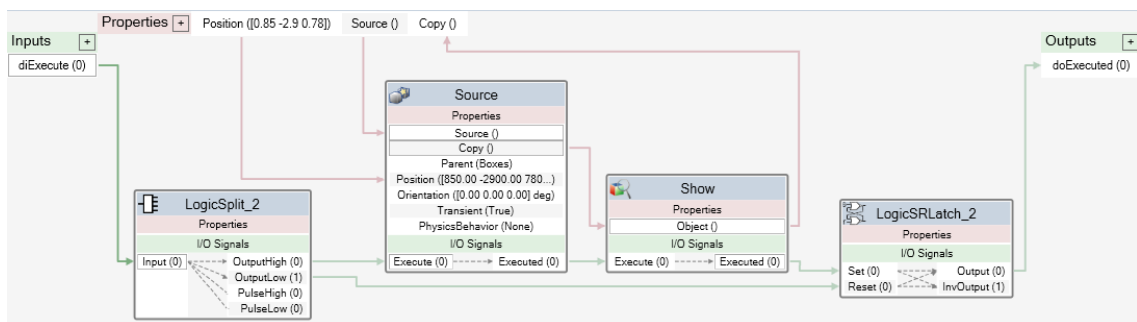
Komponenta rovněž zahrnuje simulaci činnosti dopravníku. Ta je realizována prvkem LinearMover, který umožňuje kontinuálně pohybovat s vybraným objektem zadanou rychlostí a směrem. Jako objekt je vybrán prvek Queue. Po aktivaci se tedy začnou pohybovat všechny vytvořené kopie krabice. Podoba realizované inteligentní komponenty je na obr. 5.4.



(a) Hlavní prvek Box.



(b) Část Source pro změnu velikosti a polohy zdrojového objektu.



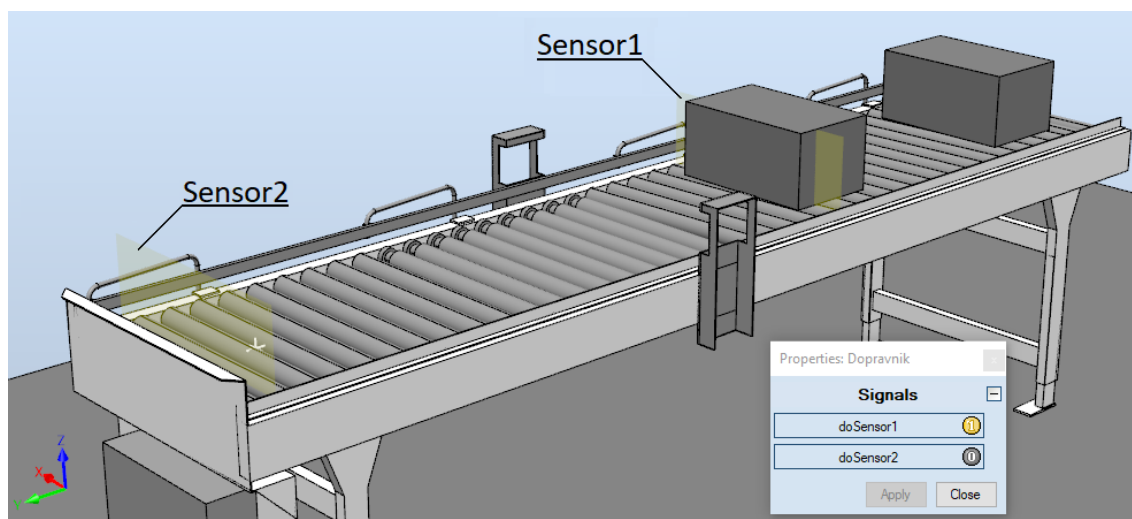
(c) Část Copy pro vytváření kopií.

Obr. 5.4: Způsob realizace inteligentní komponenty pro generování krabic.

Poslední vstup diDequeue umožňuje odebrat první prvek z fronty, aniž by došlo k jeho smazání. Této funkce je využito především při odebírání krabic z dopravníku. Jak již bylo řečeno, krabice na dopravníku jsou uloženy ve frontě. Operace provedené nad frontou, například lineární pohyb, se projeví na všech jejích prvcích. Toto chování však není žádoucí, pokud je některá z krabic z dopravníku odebrána. Vstupem diDequeue dojde prakticky k oddělení krabice od fronty. Způsob uchopení takto osamocené krabice chapadlem je popsán v kap. 5.5.

5.4 Dopravník

K dopravování krabic slouží válečkový dopravník o šířce 600 mm. Model dopravníku je součástí standardní knihovny RobotStudia. Dopravník je doplněn o dva prvky PlaneSensor, které umožňují vyhodnotit přítomnost jiného objektu. Jejich umístění na dopravníku je ukázáno na obr. 5.5. Sensor1 umožňuje generovat krabice s přesným rozestupem, Sensor2 je v roli koncového spínače, na jehož základě bude řízen pohyb dopravníku.



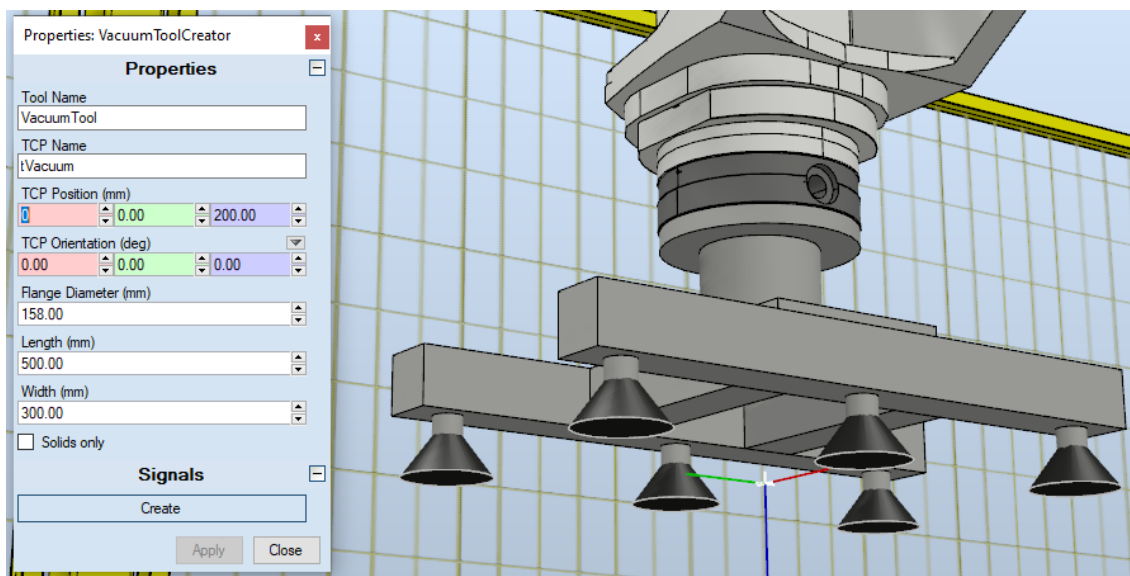
Obr. 5.5: Válečkový dopravník se dvěma snímači.

5.5 Simulace chapadla

Model chapadla byl vytvořen pomocí inteligentní komponenty VacuumToolCreator, kterou je možné získat v záložce *Add-Ins-RobotApps*. Jedná se o komponentu vytvořenou a sdílenou jiným uživatelem RobotStudia.

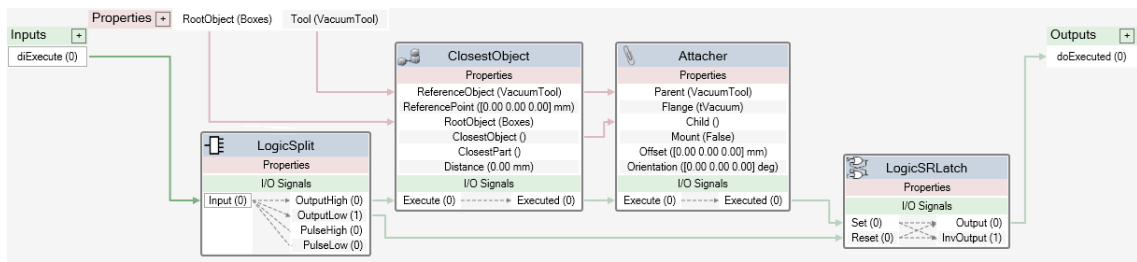
Komponenta umožňuje specifikovat název chapadla, název proměnné typu tool-data, která uchovává parametry nástroje: pozici a orientaci středu nástroje (TCP) vzhledem k přírubě robota, hmotnost, polohu těžiště, moment setrvačnosti. Tyto parametry jsou zohledňovány při pohybu robota.

Dále je možné specifikovat průměr příruby FlangeDiameter, délku Length a šířku Width chapadla. Po stisknutí tlačítka Create je vytvořen model chapadla a také definice jeho vlastností, kterou lze použít při vytváření pohybových procedur.

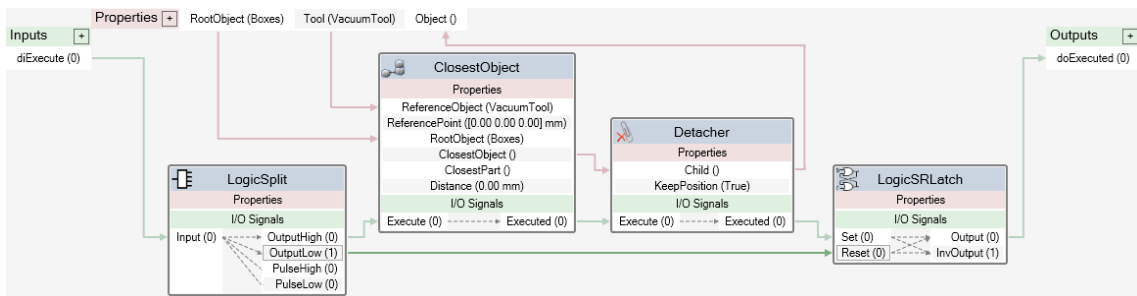


Obr. 5.6: Parametry inteligentní komponenty pro vytvoření modelu chapadla.

Samotný model chapadla však neumožňuje programově uchopit krabici. K tomuto účelu slouží vytvořená inteligentní komponenta Gripper. Ta disponuje dvěma funkcemi – přichycením objektu k nástroji a oddělením objektu od nástroje. K tomu jsou využity prvky Attach a Detach. Nástroj, se kterým komponenta pracuje, je specifikován jako jeden z parametrů komponenty. Objekt, který má být přichycen nebo oddělen, se určí vždy jako nejbližší objekt vzhledem k nástroji. Nalezení nejbližšího objektu umožňuje prvek ClosestObject.



(a) Část Attach pro přichycení objektu k chapadlu.



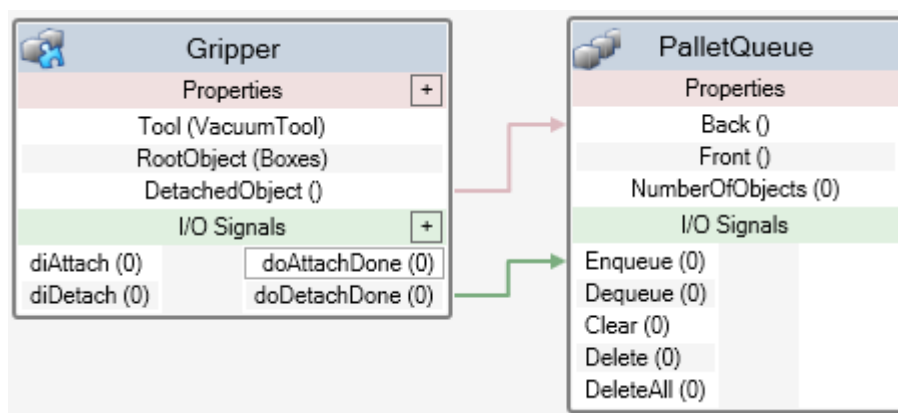
(b) Část Detach pro oddělení objektu od chapadla.

Obr. 5.7: Způsob realizace inteligentní komponenty Gripper pro simulaci chapadla.

5.6 Simulace palety

Objekty na paletě jsou uchovány ve frontě PalletQueue. S obsahem fronty lze pracovat jako s jedním objektem. Díky tomu lze jednoduše simulovat proces výměny palety. Například po dokončení nakládání může být smazán obsah palety smazáním všech prvků fronty vstupem DeleteAll.

Objekt je do fronty přidán ve chvíli, kdy dojde k jeho oddělení od chapadla. Je zde tedy propojení mezi inteligentními komponentami Gripper a PalletQueue. Způsob propojení ukazuje obr. 5.8.



Obr. 5.8: Způsob propojení inteligentních komponent Gripper a PalletQueue.

Aktivací vstupu `diDetach` komponenty `Gripper` se provede oddělení objektu od chapadla. Dokončení operace je signalizováno výstupem `doDetachDone`, který je přiveden na vstup `Enqueue` komponenty `PalletQueue`. Zároveň je oddělený objekt vystaven v parametru `DetachedObject`. Aktivací vstupu `Enqueue` dojde k zařazení tohoto objektu na konec fronty.

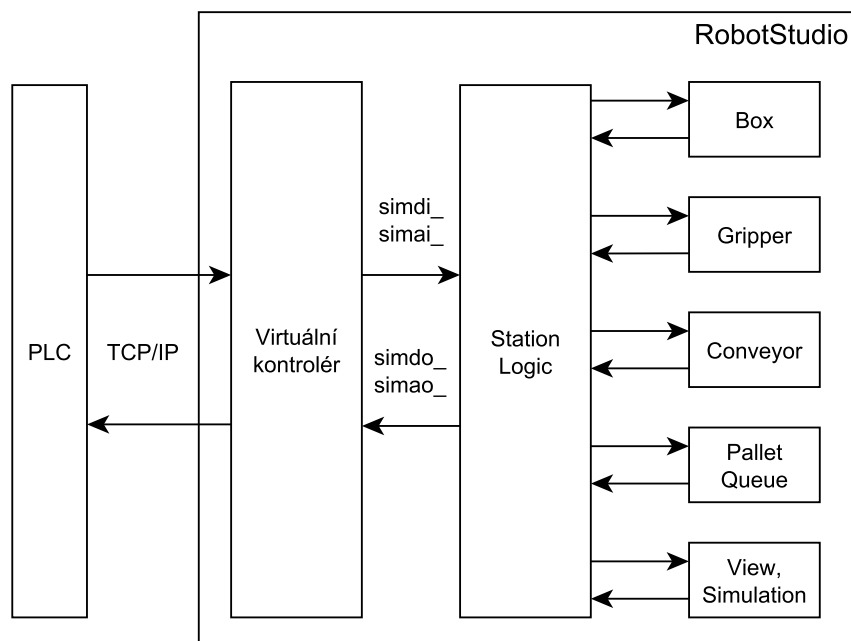
5.7 Propojení pracoviště s PLC

U reálného pracoviště jsou signály ze snímačů, např. optických bran, přivedeny na vstupní karty PLC a přes výstupní karty PLC ovládá akční prvky – pohyb dopravníku, sevření chapadla atd.

Pro účely simulace bylo potřeba vymyslet způsob, jak propojit snímače a akční prvky virtuálního pracoviště se simulovaným PLC. Po vzoru komunikace robota a PLC jsou potřebné signály přenášeny pomocí TCP/IP socketů. Komunikaci ze strany PLC zajišťuje funkční blok `RBTemplateCommTcp` popsany v kap. 2.2.1. Navržený způsob komunikace mezi PLC a RobotStudiem znázorňuje obr. 5.9.

V prostředí RobotStudio je pro účely komunikace použit virtuální kontrolér robota. O příjem, zpracování a odeslání signálů se stará samostatný program (task) `T_VIRTUAL`. Přijaté informace se přepíší do vytvořených signálů virtuálního kontroléru. Tyto je pak možné propojit s inteligentními komponentami pomocí záložky `Station Logic`.

Kromě signálů akčních prvků a snímačů, které by u reálného pracoviště byly přivedeny na některou ze vstupních či výstupních karet, obsahuje struktura navíc signály pro řízení simulace. Tab. 5.1 uvádí „příkazy“ přenášené z PLC do virtuálního pracoviště, tab. 5.2 stavové informace přenášené opačným směrem, tj. z virtuálního pracoviště do PLC.



Obr. 5.9: Znázornění způsobu přenosu signálů mezi PLC a RobotStudiem.

Tab. 5.1: Signály přenášené mezi PLC a virtuálním pracovištěm – příkazy.

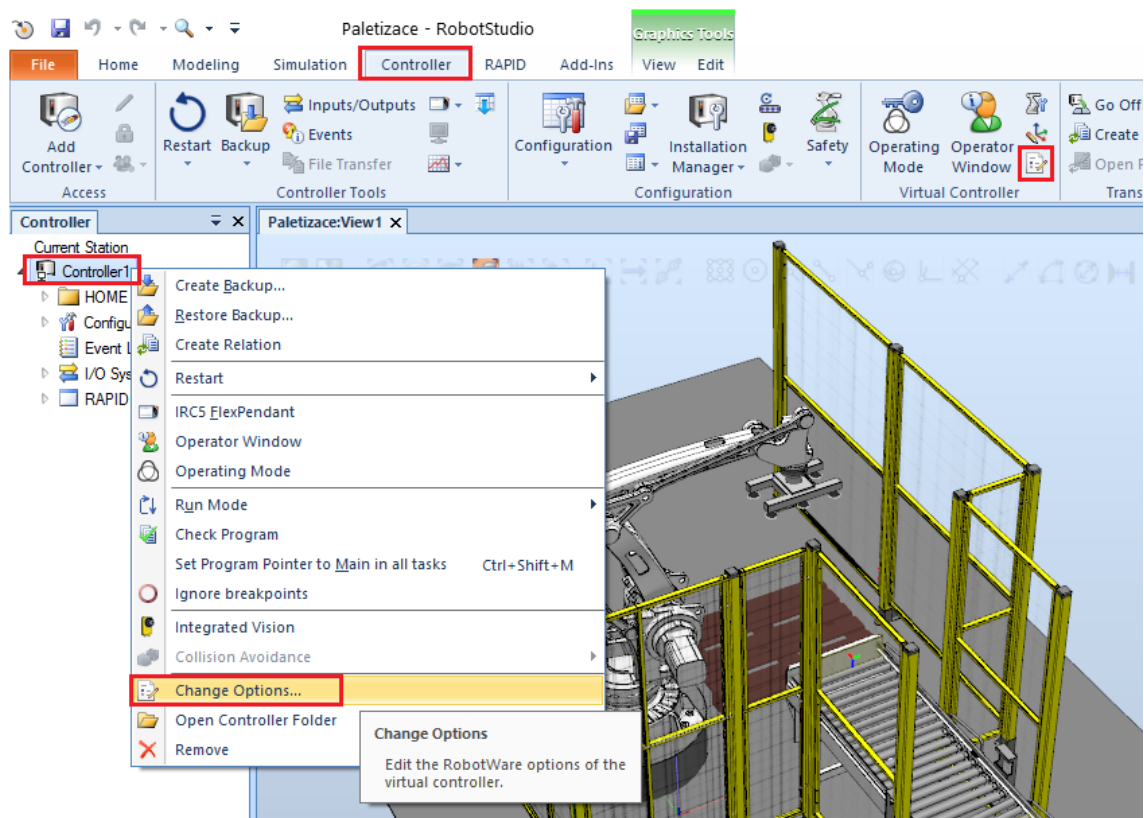
Proměnná	Datový typ	Popis
BoxCopy	BOOL	vytvoření kopie krabice
ConvMove	BOOL	ovládání dopravníku (zastaven / v pohybu)
Attach	BOOL	přichycení objektu k chapadlu
Detach	BOOL	oddělení objektu od chapadla
PalletClear	BOOL	smazání všech objektů na paletě
SimulationStart	BOOL	aktivace režimu simulace
SimulationStop	BOOL	zastavení režimu simulace
View1	BOOL	výběr pohledu kamery 1
View2	BOOL	výběr pohledu kamery 2
View3	BOOL	výběr pohledu kamery 3
View4	BOOL	výběr pohledu kamery 4

Tab. 5.2: Signály přenášené mezi virtuálním pracovištěm a PLC – stavové signály.

Proměnná	Datový typ	Popis
Sensor1	BOOL	snímač na dopravníku
Sensor2	BOOL	snímač na dopravníku, koncová poloha
BoxCopyDone	BOOL	vytvoření kopie krabice dokončeno
ConvMoving	BOOL	stav dopravníku (zastaven / v pohybu)
AttachDone	BOOL	přichycení objektu k chapadlu dokončeno
DetachDone	BOOL	oddělení objektu od chapadla dokončeno
PalletClearDone	BOOL	smazání všech objektů na paletě provedeno
SimulationStarted	BOOL	režim simulace aktivní
SimulationStopped	BOOL	režim simulace zastaven
View1	BOOL	pohled 1 aktivní
View2	BOOL	pohled 2 aktivní
View3	BOOL	pohled 3 aktivní
View4	BOOL	pohled 4 aktivní

5.7.1 Konfigurace stanice v RobotStudios

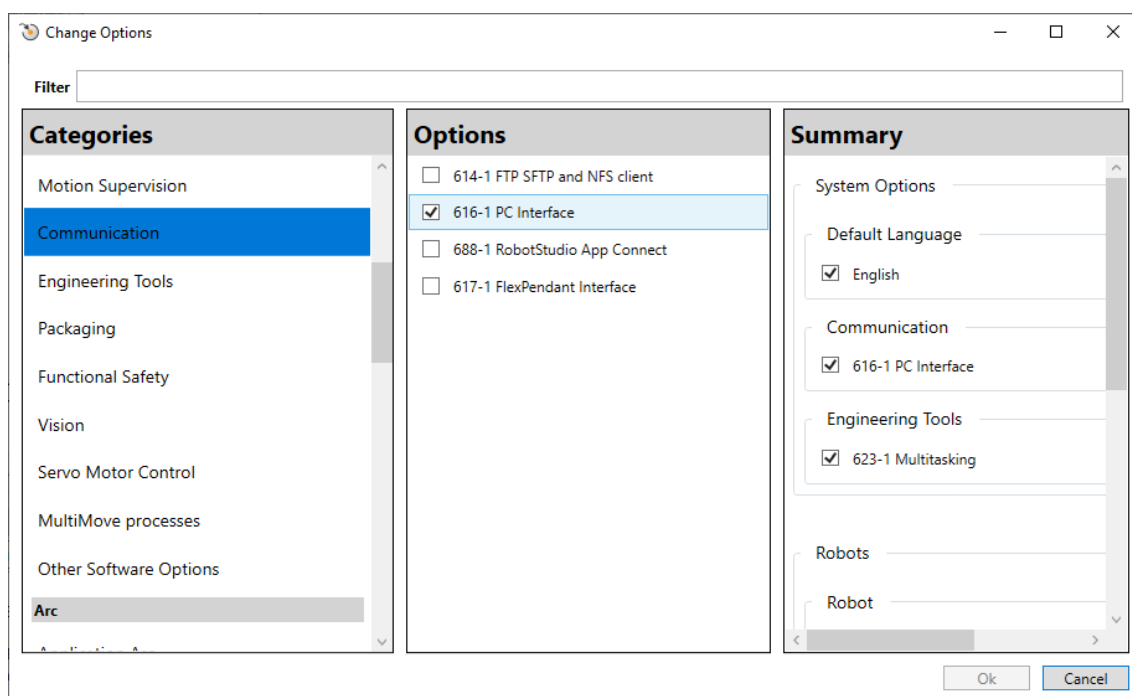
Aby bylo možné ve virtuálním kontroléru vytvářet další programy, je potřeba aktivovat rozšíření Multitasking. Kromě toho je potřeba aktivovat rozšíření PC Interface, které umožňuje mimo jiných použití funkcí pro komunikaci přes TCP/IP sockety. Obojí lze provést přes dialogové okno Change Options. Postup naznačují obr. 5.10 a 5.11.



Obr. 5.10: Možné způsoby otevření okna pro konfiguraci rozšíření.

V dialogovém okně je potřeba najít a zaškrtnout rozšíření *623-1 Multitasking* a *616-1 PC Interface*. Potvrzením změn tlačítkem Ok dojde k resetování kontroléru, všechny provedené změny týkající se kontroléru jsou vymazány. Proto je nutné ještě před změnou rozšíření vytvořit zálohu, pokud kontrolér obsahuje například vytvořené procedury či definované uživatelské signály, které mají být zachovány.

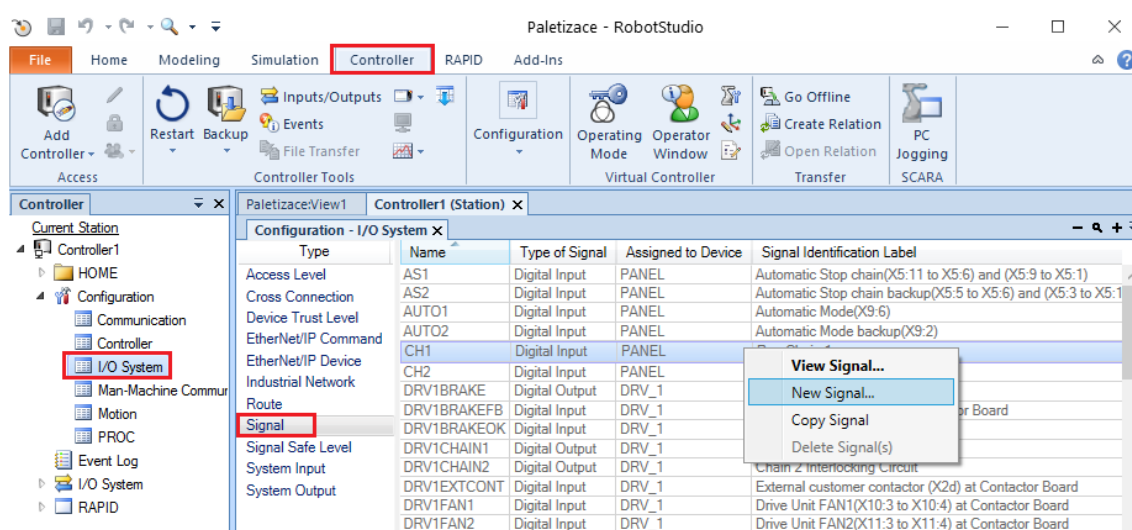
Způsob provedení zálohy se liší v závislosti na zálohovaných datech. Konfiguraci kontroléru lze uložit tlačítkem Save Parameters v záložce Controller. Proceduru lze uložit například exportováním modulu, ve kterém se daná procedura nachází. To se provede pravým kliknutím na příslušný modul v záložce RAPID a zvolením příkazu Save Module As.



Obr. 5.11: Okno pro konfiguraci rozšíření.

5.7.2 Vytvoření virtuálních signálů

Správa signálů kontroléru probíhá přes záložku *Controller-Configuration-I/O System*. Nový signál lze vytvořit pravým kliknutím na seznam a volbou příkazu New Signal. Otevřené dialogové okno slouží pro určení parametrů nového signálu: názvu, typu, přístupových práv, výchozí hodnoty aj. U analogových signálů (Analog Input, Analog Output) lze navíc určit bitovou délku a typ *signed/unsigned*.



Obr. 5.12: Okno pro správu signálů.

Potvrzením parametrů tlačítkem Ok je signál přidán do seznamu. Aby se změny projevily a signál bylo možno používat, je potřeba restartovat kontrolér.

Name	Value	Information
Name	BoxUpdate	Changed
Type of Signal	Digital Input	Changed
Assigned to Device		
Signal Identification Label	Zmena velikosti a pozice krabice	Changed
Category		
Access Level	Default	
Default Value	0	
Filter Time Passive (ms)	0	
Filter Time Active (ms)	0	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	

Value (string)
The changes will not take effect until the controller is restarted.
Minimum number of characters is <invalid>. Maximum number of characters is <invalid>.

OK Cancel

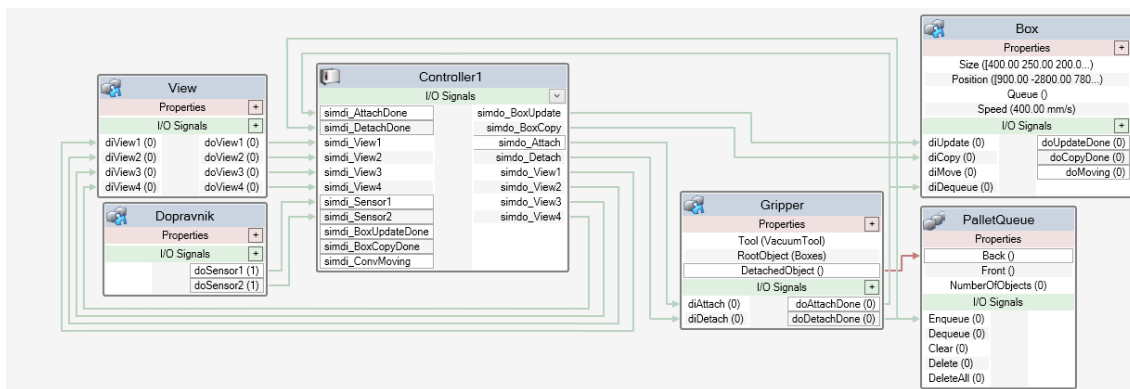
Obr. 5.13: Parametry nového signálu.

Pro zachování přehlednosti jsou názvy signálů z tab. 5.1 doplněny o předponu *simdo_*, pokud se jedná o signál typu BOOL a *simao_* v případě číselného typu. Obdobně jsou doplněny názvy signálů z tab. 5.2 o předponu *simdi_* pro signály typu BOOL, resp. *simai_* pro signály číselného typu.

5.7.3 Propojení signálů kontroléru s inteligentními komponenty

Vytvořené signály kontroléru je možné nyní propojit s inteligentními komponenty. Vazby mezi jednotlivými prvky stanice se vytváří v okně *Simulation-Station Logic*. Zde je možné vytvářet vazby buď graficky v záložce Design, nebo je spravovat pomocí tabulek v záložkách Properties and Bindings a Signals and Connections. To je výhodné zejména pokud scéna obsahuje větší množství vazeb, kdy grafický režim může být nepřehledný.

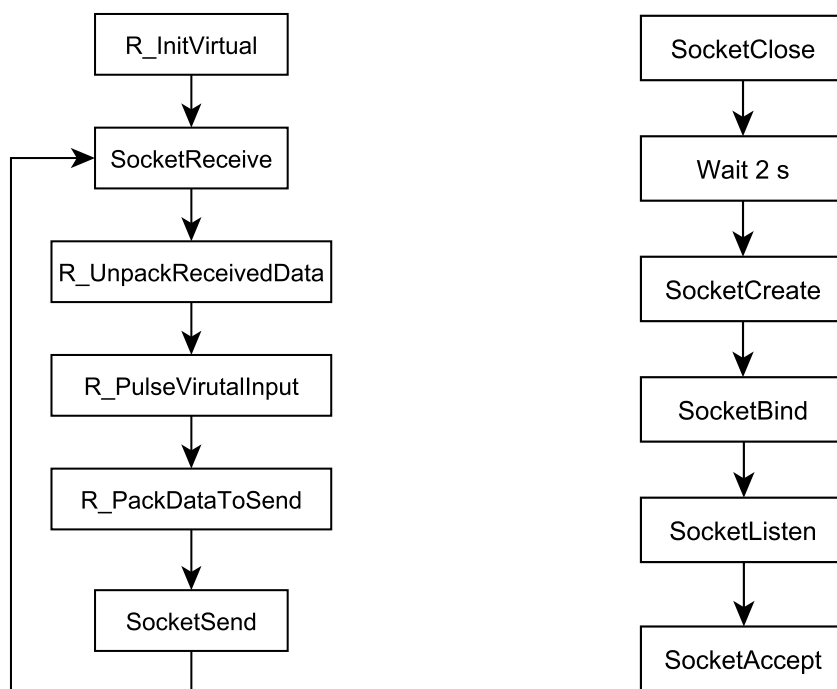
Výsledný způsob propojení ukazuje obr. 5.14. Pro zachování přehlednosti jsou některá propojení vynechána.



Obr. 5.14: Ukázka propojení signálů kontroléru s inteligentními komponenty.

5.7.4 Vytvoření programu pro komunikaci

Program *T_VIRTUAL* obsahuje dva moduly: *M_Data* a *M_Main*. Modul *M_Data* obsahuje definice konstant jako je například IP adresa a port serveru, doba čekání na připojení klienta, velikosti komunikačních struktur či pořadí signálů ve strukturách. Druhý modul *M_Main* obsahuje všechny potřebné procedury pro komunikaci s PLC. Výchozím bodem po spuštění programu je procedura main. Struktura procedury vychází z doporučení a příkladů uvedených v aplikačním manuálu ABB [27].



(a) Struktura procedury main. (b) Vytvoření TCP serveru, procedura *R_InitVirtual*.

Obr. 5.15: Posloupnost provádění programu pro komunikaci s PLC.

Procedura *R_InitVirtual* slouží pro vytvoření TCP serveru a navázání spojení s klientem. Nejdříve jsou zavřeny případně již existující sockety serveru a klienta. Následně je vytvořen socket serveru na IP adrese dané konstantou *SERVER_ADR* a portu *SERVER_PORT*. Poté program čeká na příchozí spojení. Jakmile se klient připojí, procedura *R_InitVirtual* je dokončena. Následující část programu je vykonávána cyklicky.

Instrukce *SocketReceive* čeká na příchozí data. Ta jsou ve formátu rawbytes. Jedná se o univerzální datový typ, který může být naplněn datovým typem num, byte nebo string. Maximální délka proměnné typu rawbytes je omezena na 1024 bajty. Procedura *R_UnpackReceivedData* převede příchozí zprávu na signály uvedené v tab. 5.1. K převodu rawbytes na jiný datový typ slouží instrukce *UnpackRawBytes*.

Následuje odeslání signálů z tab. 5.2. Instrukce *SocketSend* rovněž pracuje s datovým typem rawbytes. Převod signálů umožňuje instrukce *PackRawBytes*. Její volání je provedeno v proceduře *R_PackDataToSend*.

V případě, že při vykonávání programu dojde k chybě, je aktuálně prováděný cyklus ukončen a program je restartován.

6 Použití knihovny pro generování vzorů

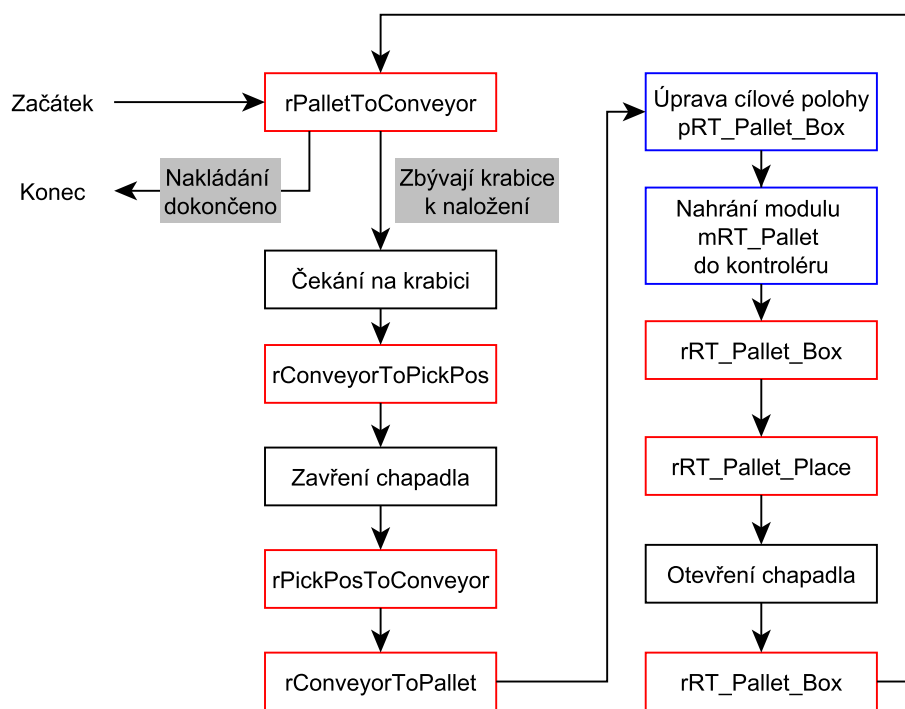
Tato kapitola popisuje způsob použití vytvořené knihovny RTPallet pro návrh paletizačních vzorů společně s knihovnou ROBOTemplate pro ovládání robota. Funkce knihovny jsou ověřeny na vytvořeném virtuálním paletizačním pracovišti.

6.1 Funkční blok pro návrh paletizačního vzoru

Použití knihovny RTPallet značně usnadňuje funkční blok RTPalletBasic. Postup pro návrh rozmístění krabic na paletě je následující. Nejdříve jsou zadány parametry palety a krabice a funkční blok je spuštěn vstupem Enable. Následně je vstupem AddLayer vytvořen požadovaný počet vrstev na paletě. Podobu vrstev lze částečně ovlivnit zvolenou metodou návrhu. Na výstupu bloku jsou vystaveny aktuální informace o vytvořeném rozmístění – počet vrstev a také počet a seznam pozic krabic. Jakmile je návrh rozmístění krabic na paletě hotov, jsou pozice krabic převedeny na pozice robota, resp. jeho nástroje, pomocí vstupu GetRobtargets. Před převodem je nutné správně vyplnit příslušné parametry – způsob uchycení krabice, výchozí natočení nástroje vzhledem k paletě – tak, aby jejich hodnoty odpovídaly robotu, pro který jsou polohy generovány.

6.2 Pracovní cyklus paletizačního pracoviště

Jakmile je znám seznam poloh pro požadované rozmístění krabic na paletě, je možné zahájit nakládání. To je realizováno sérií pohybových procedur robota. Série úkonů pro naložení jedné krabice na paletu je označena jako pracovní cyklus. Pro konkrétní virtuální paletizační pracoviště navržené v kap. 5 je navržen pracovní cyklus podle obr. 6.1. Červené orámování znázorňuje spuštění procedury s uvedeným názvem. Význam procedur popisuje tab. 6.1. Procedury jsou spouštěny z PLC pomocí funkčního bloku RBTemplateRobotCtrl.



Obr. 6.1: Pracovní cyklus paletizačního pracoviště.

Vytvořené procedury se nemění. Dosažení správné cílové polohy pro každou krabici je realizováno změnou cílového bodu pRT_Pallet_Box v předpřipraveném modulu mRT_Pallet. Úpravu bodu umožňuje funkční blok RBTemplateTargetEditor. Modul je po každé změně přenesen do PLC a procedury tedy pracují s aktuální polohou.

Tab. 6.1: Popis procedur paletizačního pracoviště

Název	Popis
rPalletToConveyor	přesun mezi paletou a dopravníkem
rConveyorToPickPos	přesun do pozice pro uchopení krabice
rPickPosToConveyor	zvednutí krabice nad dopravník
rConveyorToPallet	přesun mezi dopravníkem a paletou
rRT_Pallet_Box	přesun nad paletou na cílové umístění krabice
rRT_Pallet_Place	položení krabice na paletu (změna v ose Z)

Kromě ovládání robota je potřeba zajistit chod ostatních prvků pracoviště. Ty jsou simulovány pomocí inteligentních komponent v RobotStudiu. Jejich propojení s PLC je popsáno v kap. 5.7. Přenášené signály jsou uvedeny v 5.1 a 5.2.

Chod dopravníku je odvozen od hodnoty proměnné Sensor2, která představuje koncový doraz na dopravníku. Je-li aktivní, nachází se na odběrném místě krabice

a dopravník je zastaven. V opačném případě je dopravník spuštěn. Pohyb dopravníku je ovládán proměnnou ConvMove.

Krabice na dopravníku jsou generovány aktivací signálu BoxCopy. Nová krabice je vytvořena vždy, když je aktivní signál Sensor1 a zároveň je v chodu dopravník. Zavření a otevření chapadla je provedeno pomocí signálů Attach, Detach.

6.3 Vizualizace v mapp View

Mapp View je označení pro nástroj, který umožňuje vytvářet obrazovky pro vizualizaci a ovládání zařízení či procesu¹. Obrazovky jsou postaveny na webových technologiích (HTML, CSS, JS), nicméně jejich tvůrce webové technologie znát nemusí, jelikož obrazovky jsou vytvářeny pomocí *widgetů* formou drag and drop.

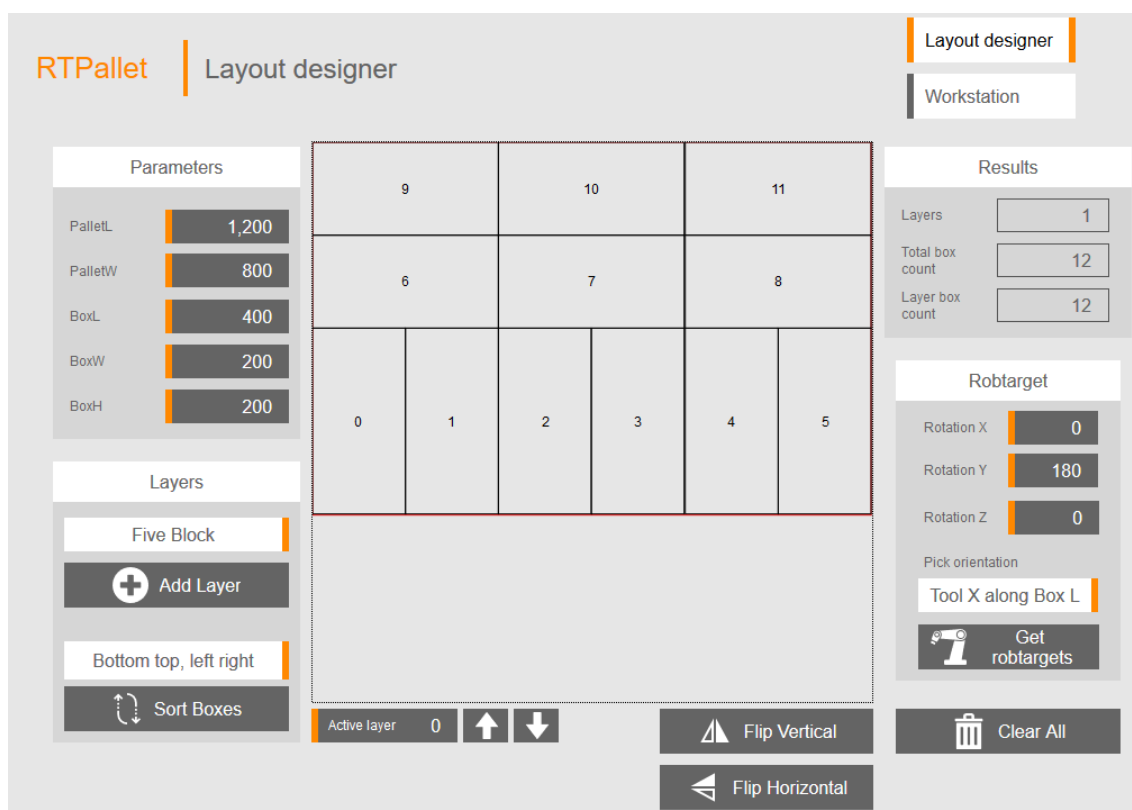


Obr. 6.2: Obrazovky jsou v mapp View vytvářeny pomocí *widgetů*. Převzato z [28].

6.3.1 Návrh rozmístění krabic na paletě

Stránka Layout designer slouží pro návrh rozmístění krabic na paletě. V zásadě se jedná o grafické rozhraní pro funkční blok RTPalletBasic. Rozvržení stránky je na obr. 6.3. Sloupec v levé části slouží pro zadávání parametrů. Mezi parametry patří rozměry palety, rozměry krabice a metoda pro návrh rozmístění. Tlačítkem Add Layer lze přidávat vrstvy, tlačítko Sort Boxes slouží pro změnu pořadí nakládání krabic ve všech vrstvách podle zvoleného způsobu.

¹HMI, Human Machine Interface



Obr. 6.3: Stránka pro návrh rozmístění krabic na paletě.

Střed stránky zabírá widget Paper, který umožňuje vykreslovat formát SVG. Zde se červenou čarou vykresluje okraje palety a černou čarou rozmístění krabic ve vybrané vrstvě. Uprostřed každé krabice je vypsáno pořadí, ve kterém bude krabice naložena. Výběr vrstvy se provádí buď zadáním čísla vrstvy do pole Active layer pod widgetem Paper, nebo pomocí šipek.

Napravo od tlačítek pro výběr vrstev jsou tlačítka Flip Vertical a Flip Horizontal, která umožňují převrátit aktivní vrstvu.

Pravá část stránky poskytuje informace o vygenerovaném rozmístění: počet vrstev, celkový počet krabic, počet krabic ve zvolené vrstvě. Panel Robtarget obsahuje pole pro zadání natočení nástroje vůči souřadnému systému palety a natočení krabice při uchopení. Tlačítko Get robtargets převede navržené rozmístění na seznam koncových poloh robota.

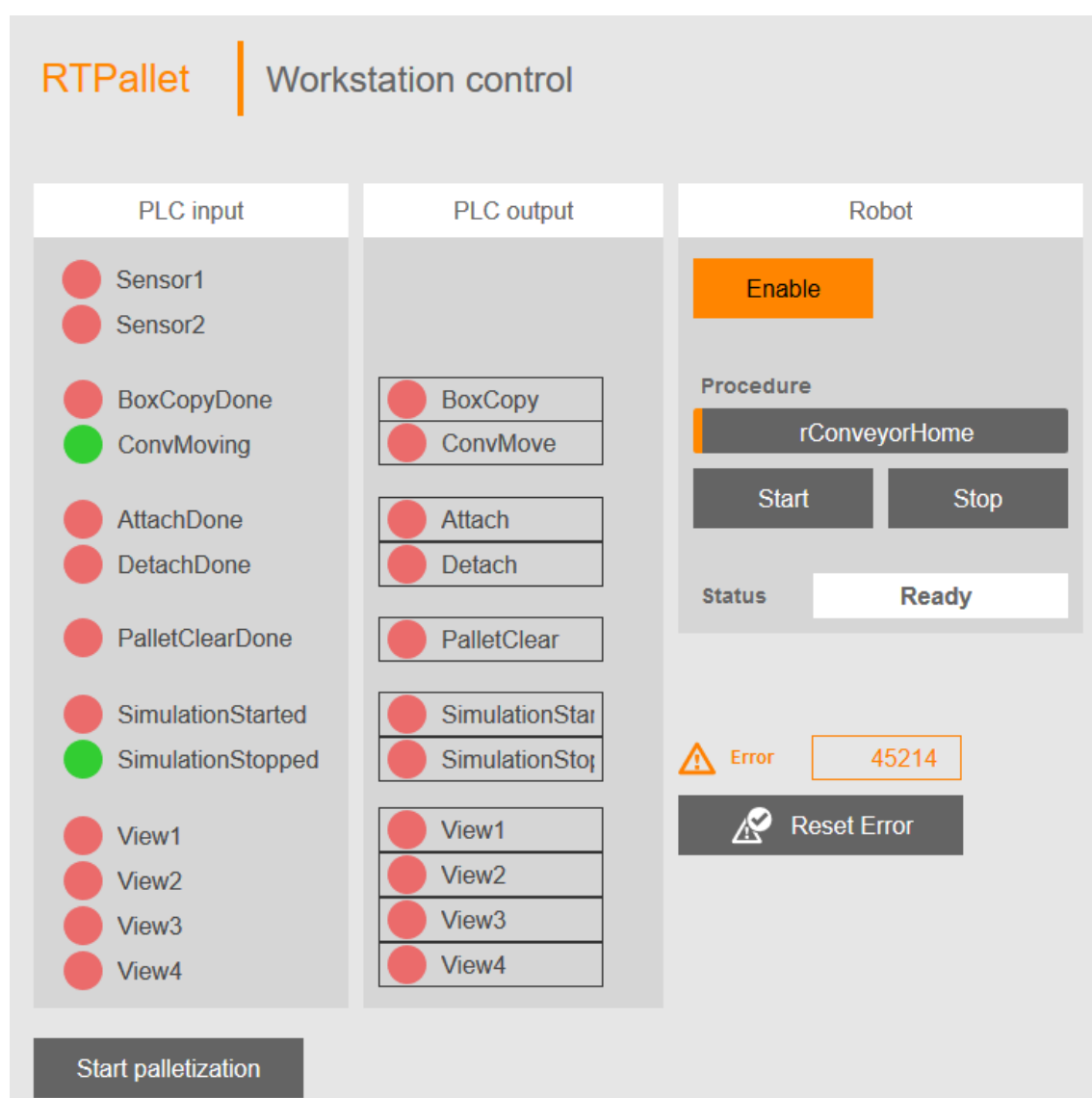
Přenos dat mezi programem PLC a vizualizací

Veškerá data mezi vizualizací a programem PLC jsou přenášena pomocí rozhraní OPC UA. Spojení mezi stránkou Layout designer a funkčním blokem RTPallet-Basic zajišťuje struktura RTPalletBasicInterfaceType. Její součástí jsou proměnné

pro zápis vstupních parametrů pro generování rozmístění – rozměry palety a krabice, metodu návrhu, způsob řazení krabic –, proměnné pro zápis hodnot tlačítek a dále proměnné obsahující informace o vytvořeném rozmístění – počet vrstev, počet krabic, SVG řetězec pro vykreslování. Na straně mapp View jsou prvky struktury připojeny k příslušným widgetům, na straně PLC k rozhraní funkčního bloku RT-PalletBasic.

6.3.2 Ovládání paletizačního pracoviště

Prostřednictvím stránky Workstation control lze ovládat virtuální paletizační pracoviště. Stránka je rozdělena na čtyři panely.



Obr. 6.4: Stránka ovládání paletizačního pracoviště.

Indikace vstupů a ovládání výstupů PLC

Sloupec PLC input obsahuje indikátory vstupních signálů PLC. Zelená barva označuje hodnotu **TRUE**, červená hodnotu **FALSE**. Sloupec PLC output umožňuje spínat výstupní signály PLC a ovládat tak jednotlivé prvky paletizačního pracoviště, například pohyb dopravníku, či přepínat mezi předdefinovanými pohledy.

Ovládání robota

Pravá část stránky obsahuje panel Robot pro ovládání robota. Tlačítko Enable zajistí, pokud je aktivní, zapnutí motorů a spuštění hlavního programu. V opačném případě je hlavní program zastaven a motory jsou vypnuty.

Tlačítkem Start lze spouštět procedury. Název procedur se uvádí do pole nad tlačítkem. Indikátor Status zobrazuje stav robota. Rozlišují se čtyři stavy:

- Motors Off, pokud jsou motory vypnuty,
- Motors On, při zapnutých motorech a zastaveném hlavním programu,
- Ready, kdy je robot připraven na vykonání procedury,
- Running, v průběhu vykonávání procedury.

Spouštění paletizačního cyklu

Pod panelem ovládání robota se nachází tlačítka ovládání paletizačního cyklu. Stisknutím tlačítka Start je zahájen automatický cyklus nakládání na paletu. V průběhu paletizace jsou výstupy PLC a robot řízeny programově. Vstupními daty pro paletizaci je seznam koncových poloh vygenerovaný na stránce Layout designer.

V případě výskytu jakékoliv chyby je zobrazen její kód společně s tlačítkem pro potvrzení a pokus o resetování.

6.4 Ověření funkčnosti

Cílem ověření funkčnosti bylo především zjistit, zdali jsou správně transformovány pozice krabic na koncové polohy robota. Ověření proběhlo na vytvořeném virtuálním pracovišti.

Při testování byla použita paleta typu EUR2 o rozměrech 1200×800 mm a krabice o rozměrech 400×200×200 mm. Rozmístění na paletě je tvořeno dvěma vrstvami (obr. 6.5). Obě byly vygenerovány pomocí metody dělení na pět oblastí. Druhá vrstva byla překlopena horizontálně. Krabice jsou z dopravníku uchopeny jejich delší hranou podél osy X nástroje. Nakládání probíhá z levého spodního rohu směrem doprava a nahoru.

9		10		11	
6		7		8	
0	1	2	3	4	5

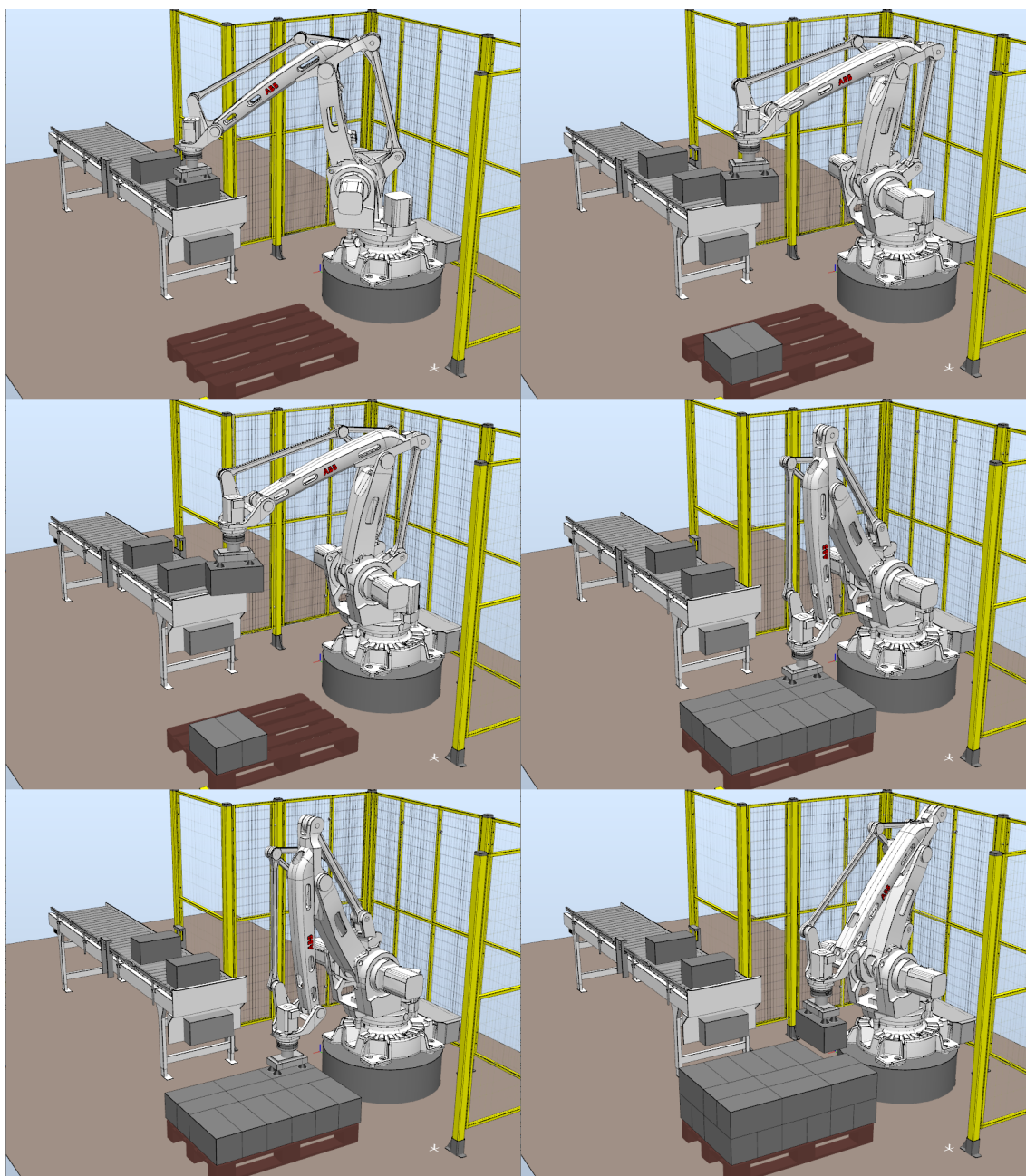
(a) První vrstva.

18	19	20	21	22	23
15		16		17	
12		13		14	

(b) Druhá vrstva.

Obr. 6.5: Vygenerované rozmístění krabic v testovacích vrstvách.

Průběh paletizace znázorňuje obr. 6.6. Je zřejmá shoda s vygenerovaným vzorem. Vytvořená knihovna pro návrh paletizačních vzorů je funkční a připravena k ověření na reálném pracovišti.



Obr. 6.6: Průběh paletizace.

Závěr

Práce je uvedena pojednáním o paletizaci, ve kterém jsou vyčleněny základní typy paletizačních zařízení včetně jejich ukázek. Větší pozornost je věnována robotickým paletizačním buňkám.

Dále je provedeno seznámení s knihovnou RoboTEMPLATE, která umožňuje řízení robota ABB z PLC společnosti B&R Automatizace. Poznatky zde uvedené jsou stěžejní pro návrh řízení automatické paletizace v praktické části, zejména způsob činnosti funkčního bloku RBTemplateRobotCtrl pro nahrávání a spouštění procedur.

Kap. 3 se věnuje metodám návrhu paletizačních vzorů, konkrétně způsobu uložení stejně velkých krabic na paletu. Z dostupných přístupů k řešení jsou vybrány metody založené na blokové heuristice, jelikož této skupině metod je v literatuře věnována značná pozornost, včetně příkladů implementace a srovnání výsledků s jinými přístupy.

Následující části jsou ryze praktické. V rámci diplomové práce je realizována knihovna, která umožňuje generovat paletizační vzory podle zadání. Knihovna je určena pro PLC společnosti B&R Automatizace. Z metod pro návrh paletizačních vzorů popsaných v kap. 3 je implementována nerekurzivní metoda dělení na pět oblastí. Jelikož k volání programů v PLC dochází v přesných časových intervalech, nejsou metody rekurzivního dělení na obdélníkové nebo L oblasti pro použití v PLC vhodné. Doba jejich výpočtu může snadno překročit nastavený interval i povolené tolerance a aktivovat bezpečnostní funkce (watchdog).

Vytvořená knihovna obsahuje samostatné funkce, které odpovídají jednotlivým krokům návrhu paletizačního vzoru. Výsledky jsou mezi funkcemi předávány prostřednictvím vhodně navržených datových struktur. Modulární koncepce knihovny umožňuje její snadné rozšíření. V rámci B&R PLC není možné dynamicky přidělovat paměť. Vzhledem k charakteru úlohy, kdy není předem znám počet umístěných krabic, bylo potřeba ošetřit možnost přetečení datových struktur pro uchovávání výsledků. Vytvořená knihovna je použita v ukázkovém projektu včetně vizualizace v mapp View, která umožňuje zadávat parametry a vizualizovat výsledky návrhu.

V rámci diplomové práce proběhlo ověření knihovny na virtuálním paletizačním pracovišti, které bylo nutné nejdříve vytvořit. Postup tvorby virtuálního pracoviště v prostředí RobotStudio popisuje kap. 5.

Pro konkrétní rozměry palety a krabice byly ve vizualizaci mapp View vygenerovány vrstvy na paletě dostupnými metodami návrhu. Následně bylo provedeno naložení krabic na paletu podle vytvořeného rozmístění. Naložená paleta byla srovnána se vzorem. Na základě provedených testů lze konstatovat, že knihovna je připravena k nasazení na reálném pracovišti.

Literatura

- [1] BIX, Laura, Javier DE LA FUENTE, Sundar RAGHAV PRASHANT, et al. Pa through Ph. KIT, L. Yam. *The Wiley Encyclopedia of Packaging Technology* [online]. Third Edition. John Wiley, 2010, s. 851-957 [cit. 2020-03-26]. DOI: 10.1002/9780470541395.ch15. ISBN 9780470541395. Dostupné z: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470541395.ch15>>
- [2] Box Palletizer VPM-BL. In: *Verbruggen: Palletizing Solutions* [online]. Emmeloord: Verbruggen Palletizing Solutions, c2019 [cit. 2020-03-25]. Dostupné z: <<https://www.verbruggen-palletizing.com/en/bl-2/>>
- [3] Robotické paletizátory. *Scott: Paletizace, interní logistika a automatizace skladů* [online]. [cit. 2020-03-26]. Dostupné z: <<https://www.alvey.cz/paletizacni-systemy/roboticke-paletizatory/>>
- [4] Obratnější robotická paletizace. FANUC: The Factory Automation Company [online]. [cit. 2020-03-26]. Dostupné z: <<https://www.fanuc.eu/cz/cs/aplikace/paletizace>>
- [5] PalVite 200 & PR Compact. *Sidel: Performance through Understanding* [online]. [cit. 2020-03-26]. Dostupné z: <<https://www.sidel.com/en/palletising-depalletising/palletization/compact-robotic-palletizer-pd-433>>
- [6] Boxes and Cartons: Pallet Maker 1-1-M. In: *Automated Manufacturing, Packaging and Logistics* [online]. Sydney: Robotic Automation P/L, c2014-2020 [cit. 2020-03-12]. Dostupné z: <<http://www.roboticautomation.com.au/robotics/standard-cells-positioners/palletising-cells>>
- [7] PanPac Standard Palletizing Cell. In: *PanPac Engineering a/s* [online]. Pandrup: PanPac Engineering a/s [cit. 2020-03-25]. Dostupné z: <<https://www.panpac.dk/standard-palletizing-cell.aspx>>
- [8] Case Palletizing Cell. In: *Omni Metalcraft Corp.: Quality Material Handling Equipment* [online]. c2020 [cit. 2020-03-26]. Dostupné z: <<https://omni.com/product-resources/application-examples/>>
- [9] ABB. *Flex Gripper Family: Grippers for palletizing – Clamp, Claw, Vacuum*. Curych, 2018. Dostupné také z: <<https://search-ext.abb.com/library/Download.aspx?DocumentID=9AKK105408A0210&LanguageCode=en&DocumentPartId=&Action=Launch>>

- [10] Lightweight gripper enables high throughput during palletizing. In: *Schunk: Superior Clamping and Gripping* [online]. SCHUNK GmbH & Co., c2020, 2016 [cit. 2020-03-28]. Dostupné z: <https://schunk.com/cz_en/news/highlights/notifications/article/2814-lightweight-gripper-enables-high-throughput-during-palletizing/>
- [11] 8011 KIT-UR-V Vacuum Gripper for Universal Robots. In: *End of Arm Tooling Products* [online]. EMI Corporation, c2020 [cit. 2020-03-28]. Dostupné z: <<https://www.emicorp.com/item/KIT%2BUR%2BV/8011-KIT-UR-V-Vacuum-Gripper-for-Universal-Robots/>>
- [12] *IRC5: Industrial Robot Controller*. ABB, 2019. Dostupné také z: <<https://search-ext.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch>>
- [13] *Technical reference manual: RAPID kernel*. ABB, c2018. Dostupné také z: <<https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC050946%20TRM%20RAPID%20Kernel%20RW%206-en.pdf>>
- [14] RAM, Balasubramanian. The pallet loading problem: A survey. In: *International Journal of Production Economics*. 1992, 28(2), s. 217-225. DOI: 10.1016/0925-5273(92)90034-5. ISSN 09255273. Dostupné také z: <<https://linkinghub.elsevier.com/retrieve/pii/0925527392900345>>
- [15] HODGSON, Thom J. *A Combined Approach to the Pallet Loading Problem: Research Report No. 81-11*. IIE Transactions [online]. August, 1981, 14(3), 175-182. DOI: 10.1080/05695558208975057. Dostupné z: <<https://www.tandfonline.com/doi/abs/10.1080/05695558208975057>>
- [16] GOMORY, R. E. a P. C. GILMORE. A Linear Programming Approach to the Cutting Stock Problem I. *Operations Research* [online]. 1961, 1961, 9(6), 849-859 [cit. 2019-10-19]. Dostupné z: <<https://www.jstor.org/stable/167051>>
- [17] DYCKHOFF, Harald. *A typology of cutting and packing problems*. European Journal of Operational Research. 1990, 44(2), s. 145-159. DOI: 10.1016/0377-2217(90)90350-K. Dostupné z: <<https://linkinghub.elsevier.com/retrieve/pii/037722179090350K>>
- [18] STEUDEL, Harold J. *Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem*. In: Management Science [online]. 1979, 25(10), s. 997-1004 [cit. 2019-10-20]. DOI: 10.1287/mnsc.25.10.997.

ISSN 0025-1909. Dostupné z: <<http://pubsonline.informs.org/doi/abs/10.1287/mnsc.25.10.997>>

- [19] BISCHOFF, E. a W. B. DOWSLAND. *An Application of the Micro to Product Design and Distribution*. The Journal of the Operational Research Society [online]. 1982, 1982, 33(3), 271-280 [cit. 2019-10-19]. DOI: 10.2307/2581493. ISSN 01605682. Dostupné z: <<https://www.jstor.org/stable/2581493>>
- [20] MORABITO, R. a S. MORALES. *A simple and effective recursive procedure for the manufacturer's pallet loading problem*. Journal of the Operational Research Society [online]. 2017, 49(8), 819-828 [cit. 2019-11-05]. DOI: 10.1057/palgrave.jors.2600588. ISSN 0160-5682. Dostupné z: <<https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2600588>>
- [21] BIRGIN, Ernesto G., Rafael D. LOBATO a Reinaldo MORABITO. *An effective recursive partitioning approach for the packing of identical rectangles in a rectangle*. Journal of the Operational Research Society. 2017, 61(2), s. 306-320. DOI: 10.1057/jors.2008.141. ISSN 0160-5682. Dostupné také z: <<https://www.tandfonline.com/doi/full/10.1057/jors.2008.141>>
- [22] LINS, L., S. LINS a R. MORABITO. *An L-approach for packing (l, w)-rectangles into rectangular and L-shaped pieces*. In: Journal of the Operational Research Society. 2003, 54(7), s. 777-789. DOI: 10.1057/palgrave.jors.2601553. ISSN 0160-5682. Dostupné také z: <<https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2601553>>
- [23] SCHEITHAUER, Guntram *Equivalence and dominance for problems of optimal packing of rectangles*.
- [24] Pallet. *Wikipedia, The Free Encyclopedia* [online]. San Francisco: Wikipedia, The Free Encyclopedia, [2001], 31 July 2004 [cit. 2020-03-11]. Dostupné z: <<https://en.wikipedia.org/wiki/Pallet>>
- [25] Digitální podnik pro výrobní průmysl: Digitální dvojče ve výrobě strojů a zařízení. Siemens [online]. Siemens, s.r.o, c1996-2020 [cit. 2020-03-29]. Dostupné z: <<https://new.siemens.com/cz/cs/products/tematicke-oblasti/digital-enterprise/discrete-industry.html>>
- [26] RobotStudio. *ABB Group, přední dodavatel digitálních technologií pro průmysl* [online]. Curych: ABB, c2020 [cit. 2020-03-10]. Dostupné z: <<https://new.abb.com/products/robotics/cs/robotstudio>>

- [27] *Application manual: Controller software IRC5*. ABB, c2014-2018. Dostupné také z: <<https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC050798%20AM%20Controller%20software%20IRC5-en.pdf>>
- [28] Mapp View: Web meets automation. In: *Perfection in Automation: B&R Industrial Automation* [online]. Eggelsberg: B&R Industrial Automation, c2020 [cit. 2020-04-06]. Dostupné z: <<https://www.br-automation.com/cs/produkty/software/mapp-view/>>

A Datové struktury funkčního bloku RBTemplateRobotCtrl

A.1 Datové struktury vstupních proměnných

Tab. A.1: Prvky struktury RBTemplateRobotCtrlInputCmdType

Název	Datový typ	Popis
MotorsOff	BOOL	zapnutí motorů
MotorsOn	BOOL	vypnutí motorů
Start	BOOL	pokračovat ve vykonávání hlavního programu
StartAtMain	BOOL	spuštění hlavního programu od začátku
Stop	BOOL	zastavení vykonávání programu (procedury)
StopAtEndInstr	BOOL	ukončení vykonávání programu po provádění instrukci
SrcFile	STRING[64]	cesta k souboru v kontroléru vzhledem k \$home adresáři, například „folder/file.mod“ vede na \$home/folder/file.mod
DstPath	STRING[64]	cesta pro uložení souboru v PLC
DstPathSim	STRING[64]	cesta pro uložení souboru v PLC, aktivní v režimu simulovaného HW
SaveFileFromController	BOOL	uložení souboru z kontroléru do PLC
RobotLicenseKey	STRING[44]	licenční klíč ROBOTemplate
ApplyRobotLicenseKey	BOOL	nahrání licenčního klíče do kontroléru

Tab. A.2: Prvky struktury RBTemplateRobotCtrlInputRobType

Název	Datový typ	Popis
ErrorReset	BOOL	potvrzení/reset chyby robota
LimitSpeed	BOOL	aktivace omezení rychlosti pohybu
AutoExecute	BOOL	provedení sekvence TransferModule, LoadModule, StartProcedure
ModuleSrcPath	STRING[64]	cesta k modulu v PLC
ModuleSrcPathSim	STRING[64]	cesta k modulu v PLC, aktivní v režimu simulovaného HW
ModuleName	STRING[36]	název modulu včetně přípony .mod
TransferModule ToController	BOOL	přenos modulu z PLC do kontroléru
LoadModule	BOOL	nahrání modulu do paměti programu robota
ProcedureName	STRING[32]	název procedury k vykonání
StartProcedure	BOOL	spuštění procedury
SpeedOverride	USINT	omezení rychlosti pohybu, hodnota 0–100 se násobí s hodnotou na FlexPendant
RelToolCords	RBTemplate RelToolType	souřadnice X, Y, Z, rotX, rotY, rotZ pro funkci RelTool
RelTool	BOOL	aktivace funkce RelTool

A.2 Datové struktury výstupních proměnných

Tab. A.3: Prvky struktury RBTemplateRobotCtrlStatusType

Název	Datový typ	Popis
Ping	BOOL	robot je dostupný v síti
Connected	BOOL	komunikace s kontrolérem aktivní
OperatorMode AutoChange	BOOL	přepnutí do automatického režimu musí být potvrzeno
AutomatOn	BOOL	kontrolér je v manuálním režimu
MotorsOn	BOOL	motory jsou zapnuty
MotorsOff	BOOL	motory jsou vypnuty
ExecutingMotionTask	BOOL	pohybový task robota běží/neběží
EmStop	BOOL	nouzové zastavení robota bylo aktivováno
ExeError	BOOL	nouzové zastavení robota bylo aktivováno
RunChainOk	BOOL	všechny bezpečnostní okruhy jsou uzavřeny
ESChainOk	BOOL	Emergency Stop okruh uzavřen
ASChainOk	BOOL	Auto Stop okruh uzavřen
GSChainOk	BOOL	General Stop okruh uzavřen
SSChainOk	BOOL	Superior Stop okruh uzavřen
SafeMoveViolated	BOOL	porušení některého z pravidel SafeMove
WorldZone	BOOL[20]	signalizace zón, ve kterých se nachází některý z robotů
SaveFileFrom ControllerDone	BOOL	přenos souboru z kontroléru do PLC dokončen
ApplyRobot LicenseDone	BOOL	nahrání licenčního klíče do kontroléru dokončeno

Tab. A.4: Prvky struktury RBTemplateRobotCtrlRobInfoType

Název	Datový typ	Popis
Active	BOOL	robot je připojen ke kontroléru
Error	BOOL	signalizace chyby
ErrorCode	BOOL	kód chyby
LimitSpeed	BOOL	omezení rychlosti pohybu aktivní
MechUnitNotMove	BOOL	signalizace pohybu robota
MotSupTriggered	BOOL	funkce Motion Supervision aktivována
TransferDone	BOOL	soubor přenesen z PLC do kontroléru
LoadModuleDone	BOOL	načtení modulu dokončeno
ProcExecuting	BOOL	probíhá vykonávání procedury
ProcDone	BOOL	procedura byla dokončena

Tab. A.5: Prvky struktury RBTemplateRobotMotionDataType

Název	Datový typ	Popis
X	REAL	poloha nástroje v ose X
Y	REAL	poloha nástroje v ose Y
Z	REAL	poloha nástroje v ose Z
Q1	REAL	orientace nástroje, kvaternion 1
Q2	REAL	orientace nástroje, kvaternion 2
Q3	REAL	orientace nástroje, kvaternion 3
Q4	REAL	orientace nástroje, kvaternion 4
Cf1	SINT	konfigurace os robota
Cf4	SINT	konfigurace os robota
Cf6	SINT	konfigurace os robota
Cfx	SINT	konfigurace os robota
Rax1	REAL	natočení osy 1 robota
Rax2	REAL	natočení osy 2 robota
Rax3	REAL	natočení osy 3 robota
Rax4	REAL	natočení osy 4 robota
Rax5	REAL	natočení osy 5 robota
Rax6	REAL	natočení osy 6 robota
Eaxa	REAL	poloha externí osy A
Eaxb	REAL	poloha externí osy B
Eaxc	REAL	poloha externí osy C
Eaxd	REAL	poloha externí osy D
Eaxe	REAL	poloha externí osy E
ActTCPSpeed	REAL	aktuální rychlost [mm/s]
RefTCPSpeed	REAL	referenční rychlost [mm/s]
ToolName	STRING[32]	aktivní nástroj
WobjName	STRING[32]	aktivní souřadný systém

B Obsah přiloženého CD

- Text diplomové práce ve formátu PDF.
- Projekt pro Automation Studio 4.6 se zdrojovými soubory vytvořené knihovny RTPallet, vizualizací v mapp View a příkladem použití knihovny.
- Projekt virtuálního paletizačního pracoviště pro RobotStudio 2019 včetně procedur pro ovládání robota.
- Ukázkové video s průběhem paletizace.